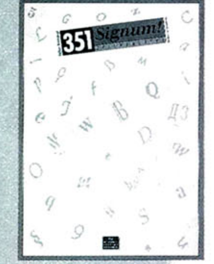

ATARI
ST
COMPUTER
Die Fachzeitschrift für den ATARI-ST Anwender.
Oktober 89
10
DM 7,- Ös. 56,- Sfr. 7,-
Netzwerke
am ST
ATARI STE
Konkurrenz für Amiga ?
Supercharger
MS-DOS-Emulator
ATARI-Messe
in Düsseldorf

Die Welt des Buches: Lesen und lesen lassen.



"Das Atari ST 1 x 1" für 49,-- DM, ISBN 3-9801834-3-2.

Autoren, Auflagen, Bestseller. Wo Themen sind, da entsteht auch immer Literatur: Der Bildband zum Film, der Film zum Buch, das Buch zur Software. Das hat sich auch unser Software-Support gedacht, denn lesen ist besser als Sprechen. Krächz. **1** Einmaleins. Klingt zwar einfach, doch wir wissen, daß beim Umgang mit dem ST Zusammenzählen nicht alles ist. Volker Ritzhaupts *"Atari ST 1 x 1"* beschreibt daher, was Anfänger und Fortgeschrittene wissen wollen und lernen sollten. Ein unentbehrliches (Nachschlage-)Werk, das auf 320 Seiten (unterstützt von über 200 Abbildungen) unbestechlich Auskunft gibt. Das Utility zum Lesen ist in angenehmer Sprache geschrieben, selbst verstörte Handbuchleser winden sich in Zuversicht. Auch das neue Betriebssystem TOS 1.4. wird hier bereits berücksichtigt. **2** *Signum!* war im letzten Jahr der erste Anlaß für ein Buch. *"Das Signum! Buch"*: Volker Ritzhaupt sagt, zeigt, erklärt alles Wissenswerte zur definitiven Textverarbeitung. Der Anfänger hat dabei erhebliche Vorteile, der Fortgeschrittene aalt sich in Tricks und Hintergrundwissen. Ein Muß für jeden *Signum!*-User. **3** Der Untertitel "Ein Buch zur Gestaltung" ist Programm für zwei großformatige Paperbacks mit den schönen Titeln *"351 Signum!-Zeichensätze"* und *"547 neue Signum!- Zeichensätze"*. Alle derzeit verfügbaren *Signum!*-Fonts sind hier versammelt, im Anhang findet der geneigte Leser lohnende Anregung für die Gestaltung von *Signum!*-Dokumenten. Viel Spaß beim Lesen!





"351 Signum!-Zeichensätze", Paperback, ISBN 3-9801834-1-6, 29,— DM und "547 neue Signum!-Zeichensätze", ISBN 3-9801834-2-4, 39,— DM.



Ein Satellit von Signum! ist "Das Signum!-Buch.": 432 Seiten, fester Einband, 59,-- DM . Im Buchhandel unter ISBN 3-9801834-0-8.

APPLICATION SYSTEMS HEIDELBERG

Die Kunst der ST Software.

Application Systems Heidelberg Software GmbH, Englerstraße 3, Postfach 10 26 46, D-6900 Heidelberg 1, Telefon (0 62 21) 30 00 02, Fax (0 62 21) 30 03 89 **in Österreich:** Reinhart Temmel Ges.m.b.H., Markt 109, A-5440 Golling, Telefon (0 62 44) 7081-0, Fax 7 18 83. **in der Schweiz:** DTZ DataTrade AG, Langstraße 94, CH-8021 Zürich, Telefon (01) 2 42 80 88, Fax 291 05 07.

EDITORIAL

Die ATARI-Messe in Düsseldorf ist nicht erst seit diesem Jahr das wichtigste Messe-Ereignis für ATARI-AnwenderInnen; alle wichtigen Firmen, die den ST unterstützen, treffen sich hier, arbeiten gezielt mit neuen Produkten auf diesen Termin hin.

Das wissen auch die AnwenderInnen. Folgerichtig sind die Stände permanent umlagert, immer wieder bilden sich Schlangen. So bleibt kaum Zeit für Gespräche über Stand-Grenzen oder für einen kleinen Messerundgang, der gerade für uns wichtig ist. Vor allem aber wird unter solchen Bedingungen ein einigermaßen entspanntes Messegespräch mit BesucherInnen schier unmöglich.

Freilich sind Stoßzeiten auf Messen völlig normal. Wenn man aber drei Tage lang von früh bis spät solchem Trubel nicht mehr entkommen kann, bleibt der einzig logische Schluß: Die Messe ist zu kurz! (ATARI hilf!)

Düsseldorf ade!

Der typisch studentische Plural ("AnwenderInnen", siehe oben) ist übrigens volle Absicht. Erstens täte der Szene eine Diskussion über die Diskriminierung von Frauen in der Computer-Sprache nicht schlecht, zweitens arbeiten viele StudentInnen mit dem ST, und drittens - sind auch die meisten Entwickler (hier ohne "Innen", um hervorzuheben, wie hoffnungslos wenig Programmiererinnen es gibt) Studenten oder Schüler.

Wer auf der Messe nur einigermaßen die Augen offenhielt, mußte das bemerken: Die meisten Soft- und Hardwareentwicklungen für den ST stammen von jungen Entwicklern, Schülern und Studenten. Einerseits zeugt das von der hohen Anziehungskraft des ST für finanzschwache Schichten, zudem auch von der jugendlichen Innovationskraft, die dem ST-Markt innewohnt.

Andererseits jedoch lauert hier eine Gefahr: Professionelle ProgrammiererInnen sind Mangelware, die meisten Produkte entstehen mehr oder weniger als "Nebenbeschäftigung". Letztlich wirkt sich das auf die Qualität aus, auch wenn man fast allen ST-Entwicklern überdurchschnittliches Engagement attestieren kann. Langfristige und umfangreiche Projekte sind so nur schwer zu verwirklichen, und die Programmpflege und AnwenderInnenbetreuung wird für beide Seiten zum Alptraum, wenn sich der studentische Entwickler während des Studiums noch damit befassen muß.

Muß man den Firmen nicht Kurzsichtigkeit vorwerfen, wenn sie statt der ausgebildeten Fachkraft doch lieber den viel billigeren Studenten engagieren? Vielleicht. Indessen: Welcher professionelle Programmierer befaßt sich denn mit dem ST? Angesichts des chronischen Arbeitskräftemangels in der Informatik können sich die AbsolventInnen der Hochschulen den Arbeitsplatz aussuchen - und wer wählt dann den Job bei einer Firma, die eine Nische auf dem immer noch wenig angesehenen ATARI-Markt beackert, wenn er auch bei Apple unterkommt? Wer will Programme schreiben, die auf dem unsteten und von Raubkopierern verdorbenen ST-Markt für hundert oder zweihundert Mark verkauft werden, wenn er für das gleiche Programm von UNIX-Anwendern mehrere Riesen einstreichen kann?

ATARI Deutschland mag die schleichende Unprofessionalität des ST-Marktes wohl erkannt haben; Messe-Initiativen wie DTP- und Netzwerkvorführungen zeugen davon. Gleichzeitig aber kommen aus Sunnyvale neue Modelle wie das Handheld-Spielchen oder der STE, die in die Computer-Kid-Ecke zielen, der portfolio als Yuppie-Spielzeug oder der an einigen Stellen engstirnig beschränkte TT (viel zu kleines und zudem mehr als gewöhnungsbedürftiges Gehäuse, zu kleine Platten-Grundausstattung, nur ein VME-Bus-Steckplatz). Bei solchen Modellen, darüber waren sich viele Entwickler auf der Messe in leiser Trauer einig, bleiben Zweifel bestehen an der Ernsthaftigkeit der ATARI-Strategie, mit TT und ATW rechtzeitig in den Workstation-Markt einzudringen.

Hoffen wir, daß ATARI nicht beim TT die Fehler der ST-Familie wiederholt, als da wären mangelnde Erweiterbarkeit (kein oder wie beim MegaST nur ein System-Steckplatz, zudem zu kleine Gehäuse), fehlende Programmierrichtlinien und fehlerhafte Betriebssystemroutinen, die ProgrammiererInnen zwingen, auf zweifelhaften Wegen um das Betriebssystem herum zu wandeln. Angesichts illustrer Konkurrenz im DTP-Bereich (Apple und Konsorten), bei den kleinen Workstations (SUN, HP, DEC...) und auch im Low-End-Bereich (Apple scheint sich zum Volks-Mac durchzuringen, und Commodore ist ja auch noch da) wird ATARI sich zusammenreißen müssen. Noch ist Zeit.

Trotz allem in großer Vorfreude auf die nächste ATARI-Messe.

Claus Brod

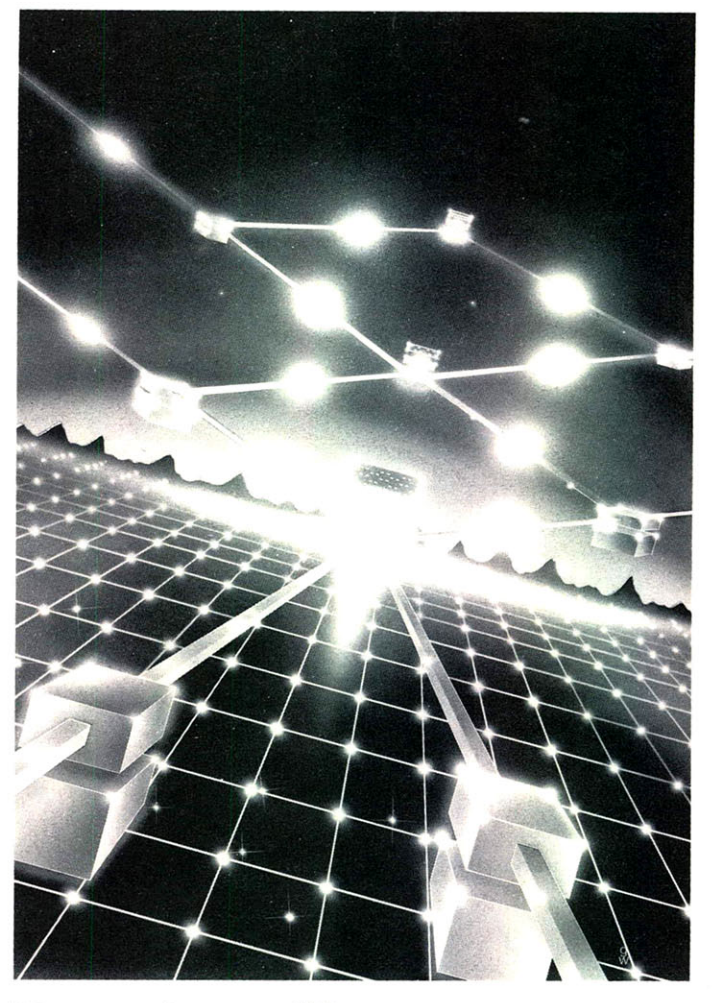
I N H A L T

SOFTWARE

HARDWARE

EMULATIONEN

GRUNDLAGEN



Netzwerke am ST

Netzwerke finden heutzutage in Betrieben, Büros und Redaktionen immer mehr Freunde. Quasi überall, wo viele auf ein und dieselben Daten zugreifen wollen, ohne daß sie sich vorher diese Daten mühsam kopieren und auf dem neuesten Stand halten müssen. Auch für den ATARI ST gibt es einige solcher Netzwerklösungen, von denen wir Ihnen in diesem Heft zwei vorstellen wollen: das eLAN- und das rho-/PAMs-Netz.

Seite 37 und 43



DFÜ-Ecke

Es ist mal wieder soweit, wir stellen Ihnen in dieser Ausgabe eine neue Rubrik vor: die DFÜ-Ecke. Hier werden wir Ihnen Neuigkeiten der "DFÜ-Szene", Grundlagen (wie XModem), Informationen und neue Mailboxen vorstellen. Sicherlich werden nicht alle von Ihnen wissen, was man mit der Datenfernübertragung alles machen kann. Aus diesem Grunde wird in dieser ersten Folge erst einmal eine kleine Einführung geboten, damit später auch alle interessierten Leser mitlesen können.

Seite 169



Düsseldorf im August
Die ATARI-Messe

Vom 25. bis 27. August fand Düsseldorf zum dritten Mal die ATARI-Show statt. Mit Sicherheit waren in erster Linie die zwei Weltpremieren die Hauptattraktion: Der ATARI TT wurde diesmal in aller Öffentlichkeit gezeigt und ebenso ein schon in absolute Vergessenheit geratenes, vor langer Zeit bereits angekündigtes ST-Modell mit besserer Grafik, der ATARI STE. Doch nicht nur ATARI hatte etwas Neues zu bieten, auch die große Anzahl an Soft- und Hardwareentwicklern zog alle Register.

Seite 6

Projekt

MGP -
Der GAL-Prommer
Programmierte Logik

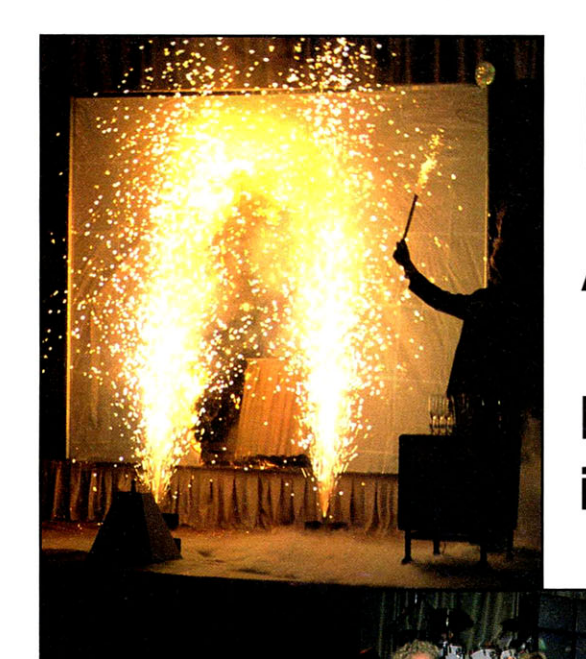
Wenn man eine logische Schaltung aufbauen und entwerfen will, kann man auf die altbewährte TTL-Logik zurückgreifen, die mittlerweile eine umfangreiche Standardbibliothek an Bausteinen mit vielen Grundfunktionen zur Verfügung stellt. Wesentlich unproblematischer läßt sich allerdings mit einem GAL arbeiten, mit dem man mehrere Standardbausteine einsparen kann. Das bringt Platz auf der Platine. GALs sind außerdem löschbar und mit einer neuen Logik zu programmieren. Unser dreiteiliges Projekt in der ST Computer bietet Ihnen die Möglichkeit, sich selbst ein GAL-Programmiergerät zu bauen. Dazu werden alle erforderlichen Unterlagen veröffentlicht. Doch diesmal wollen wir uns erst etwas mit den Grundlagen zum GAL beschäftigen.

Seite 164

PROGRAMMIERPRAXIS

AKTUELLES

RUBRIKEN



Die ATARI-Messe

Düsseldorf im August

Für ATARI-Anwender ist der Sommer seit 2 Jahren zu etwas Besonderem geworden. Nicht nur wegen der Sonne, die wir dieses Jahr reichlich gehabt haben, oder dem Urlaub, den man zu Hause auf dem Balkon mit einer ausgewachsenen Palme eines skandinavischen Möbelfabrikanten und einem exotischen Drink in der Hand verbringen konnte oder auch im Bermuda-Dreieck mit vielen Palmen, aber mit genausoviel Platz wie auf dem heimischen Balkon. Nein, nicht nur deswegen ist für den ATARI-Anwender der Sommer interessant, sondern weil sich jedes Jahr, wenn der Sommer zu Ende geht, die Tore der Düsseldorfer Messe für alle möglichen Zuschauer öffnen, die die mittlerweile schon etablierte ATARI-Show besuchen möchten.

Mit Donner und Rauch kam der ATARI TT



Jack Tramiel fühlte sich sichtlich wohl.

Und weil dieser Sommer nicht anders war als der Sommer letzten Jahres, wurde vom 25. bis 27. August in der Stadt am Rhein zum dritten Mal die ATARI-Show veranstaltet. Über 35000 Besucher gingen während dieser drei Tage auf der Suche nach Information hungrig durch beide Messehallen. Mit Sicherheit waren in erster Linie die zwei Weltpremieren die Hauptattraktion für die Besucher. Der ATARI TT wurde diesmal in aller Öffentlichkeit gezeigt und ebenso ein schon in absolute Vergessenheit geratenes, vor langer Zeit bereits angekündigtes ST-Modell mit besserer Grafik, der ATARI STE. Aber nicht damit fing alles an, sondern mit dem zuvor im Konferenzraum des Düsseldorfer Hilton.

Magic, Licht und Beine

Was haben ein holländischer Entertainer (nicht Rudi Carrell, keine Panik!), vier holländische Mädchens und der ATARI TT gemeinsam? Stimmt, gar nichts! Das ist auch der Eindruck, der nach der Präsentation der neuen Maschinen am Donnerstag vor der Messe übrig geblieben ist. Eine ungewöhnliche Show mit viel Musik und einem nicht gerade komischen Entertainer waren der Rahmen der ersten öffentlichen Vorstellung (beschränkt auf Fachleute und Händler) der zukünftigen ATARI-Produkte. Es wurde wenig geredet, sondern eher mehr gesungen und fürs Auge gab es nicht nur die zwei neuen Modelle, sondern jede Menge Laserlicht und schöne Beine. Ja, die Beine der 'Four Sisters', einer Gruppe aus Holland, die der Veranstaltung den nötigen Swing verleihen sollte.

Eine so wichtige Veranstaltung kann nicht ohne all die Leute stattfinden, die in den USA für die Entwicklung sowie Support verantwortlich sind. Jack, Leonhard und Sam Tramiel sowie eine nicht gerade kleine Mannschaft an Entwicklern waren anwesend. Nichtsdestotrotz präsentierte Alwin Stumpf die neuen Modelle als wären sie Entwicklungen aus dem gerade eben geschlossenen ATARI-Entwicklungszentrum in Braunschweig. Die Bedeutung des deutschen Marktes, weltweit angesehen, ist mit den Jahren immer mehr gestiegen. So gewinnt die Figur des Geschäftführers von ATARI Deutschland immense Bedeutung.

Nach einer guten Stunde Tanz auf dem Parkett und ebenfalls tanzenden Lichtern auf der Leinwand tauchten hinter einer Wolke von farbigem Rauch und einem ebenso farbenprächtigen Feuerwerk auf einem präparierten Hintergrund die zwei neuen Rechner von ATARI auf.

Die Lichter erloschen langsam, während die fast 300 Gäste wild applaudierten, der holländische Entertainer murmelte etwas in flämisch, während er seinen Champagner trank, und Alwin

Stumpf lächelte zufrieden auf der Bühne. Uns trieb der Rauch nach draußen. Die Show war beendet! Es lebe die Show!

Drei Tage rund um die Messe

Schon am ersten Tag um 9 Uhr, als wir zum Messegelände kamen, sammelten sich die Leute vor den Toren. Als eine Stunde später geöffnet wurde, strömten sie von allen Seiten durch die Korridore. Der erste Blickfang war mit Sicherheit der groß angekündigte ATARI TT, der nur auf dem Zentralstand von ATARI zu sehen war.

Der TT entäuschte auf den ersten Blick viele Besucher aufgrund seines Äußeren. In einem vielleicht sehr modernen Design untergebracht, sieht er so aus, als hätte man vergessen, einen Dekkel anzufertigen. Man ist von dem grauen Kasten weggekommen, stattdessen hat der TT ein wesentlich angenehmeres Creme-farbenes Gehäuse bekommen. Die Tastatur ist jedem Besitzer eines Mega ST genauso bekannt wie die Maus. Ob sich hier etwas ändern wird, war nicht zu erfahren. Da der TT eine Reihe verschiedener Grafik-Modi beherrscht, wird er von Haus aus mit einem VGA-ähnlichen Monitor geliefert. Zur Grundausstattung gehört ebenso ein interner Speicher von 2 MB, eine Festplatte mit 30 MB und das Betriebssystem *TOS 030*, das kompatibel zum ST-TOS sein soll. Von Unix war nichts zu sehen, und man muß sich bis zur Systems in München im Oktober dieses Jahres gedulden, um Genaueres zu erfahren.

Für uns war der TT keine große Neuheit mehr, da wir (wie andere Kollegen) auf der CeBIT in der Lage waren, diese Maschine zu betrachten. Neu war uns nur das Gehäuse-Design. Ebenso haben wir bereits mehrmals über die neuen Generationen von ATARI-Modellen geschrieben. Seit guten zwei Jahren wird schon über diesen Rechner gesprochen und geschrieben. Gerade deswegen ist es unbefriedigend, daß man nicht erfahren kann, wann dieser Rechner auf den Markt kommen wird, wann wir in der Redaktion ein Exemplar bekommen können, womit wir lange und in Ruhe arbeiten könnten. Durch Probieren und Experimentieren könnte man dann ein gewisses Verhältnis zu dem Gerät bekommen. Im Rahmen dieses Messeberichts und mit all diesen offenen Fragen ist es nicht sehr sinnvoll, weiter zu spekulieren und mehr darüber zu sagen.

Das andere neue Modell, der sogenannte ATARI ST^{E}, ist ein ATARI ST, der wohl hauptsächlich für den Spielemarkt gedacht ist. Über ihn ist in diesem Heft noch mehr zu erfahren. Aber nicht nur bei ATARI konnte man alles, was glänzt, bewundern. Bei den 155 Ausstellern, die auf der Messe versammelt waren, gab es eine Menge zu sehen. Setzen wir unseren Rundgang fort.

Die harten Sachen

Wie bei anderen Rechnern ist die Menge der Peripheriegeräte, die von Fremdanbietern hergestellt werden, sehr groß. Von einzelnen Floppies bis zum exotischen Emulator wird alles mögliche als Zusatz für den ST angeboten. Und weil mittlerweile das Angebot so groß geworden ist und die Ansprüche der Anwender gewachsen sind, bemühen sich fast alle Hersteller um mehr Professionalität ihrer Produkte.

Der Markt für Festplatten scheint immer noch vielen Firmen interessant genug, um sich auf dieses Feld zu wagen, obwohl auch hier schon Preiskämpfe ausgefochten werden. Darüber freut sich der Anwender, den Einkaufs- und Marketingchefs der beteiligten Firmen aber macht das bereits einige Kopfschmerzen. Wie auch immer, dies ist ein mitleidsloses Geschäft, und mit guten Ideen kann man auch noch auf einem so heiß umworbenen Markt erfolgreich sein.



Die optische Platte von protar

Die recht junge Firma **protar** - Ableger der unter Mac-Usern etwas bekannteren **formac** - etwa setzt auf Solidität: Ihre Festplattenserie (20 bis 180 MB, 998 bis 3998 DM) zeichnet sich durch leise Lüfter und schnelle Platten, ein stabiles Gehäuse und eine in GEM eingebundene Software aus. 12 Partitionen pro Platte und die Autobootfähigkeit sind wir bereits gewohnt, Schreib- und Paßwortschutz schon weniger. Besonderer Gag: Beim Hochfahren meldet der Treiber die Laufwerk-Icons des Desktops selbst an! Auf ihrer Messepremiere zeigte protar auch eine Wechselplatte für 2498 DM und sogar eine magnetooptische Platte für den ST: 620 MB faßt das 998 DM teure Medium, auf dem das eingebaute Sony-Laufwerk im Mittel innerhalb von 90 ms jede Spur findet. Die Platte kostet - halten Sie sich fest - im Moment samt Medium noch 11998 DM. Bis die Magnetooptik also für Sie und mich erschwinglich wird, kann es noch ein wenig dauern. Immerhin wurde endlich deutlich, daß diese neue Speichertechnologie die Entwicklungslabors verläßt und verfügbar wird. Vom Videoprozessor TI34061 getrieben, stellt der entspiegelte 19"-Großbildschirm von protar 1280x1024 Punkte bei einer Wiederholfrequenz von mehr als 70 Hz dar. Kostenpunkt: 4298 Mark. Alles in allem also für eine so junge Firma ein recht imposanter Einstieg.

Überhaupt wimmelte es - mehr noch als im letzten Jahr - von Großbildschirmen und Grafikkarten. Bei Binnewies & Kammler nennt sich die Karte verführerisch "NEXTscreen" und liefert auf einem Multisync 800x600 Punkte (Wiederholfrequenz 55 Hz), auf einem SM124 immerhin noch 640x480 Punkte. Die Karte kostet samt Einbau 225 Mark. Zurück zu meinen Lieblingen, den Platten: Auch **Binnewies** hat eine Festplattenreihe ("NEXT-file" - wie das Jobs wohl findet?) zu bieten (20 bis 85 MB, Preise von 1095 bis 1595 DM), SCSI-Platten allesamt, die durch eigene Software (8 Partitionen) und Hardware-Schreibschutz auffallen. Bei so vielen Platten am Stand, die mich ein wenig ängstlich anblinzelten, hätte ich beinahe die NEXTboard-Tastatur unter meinen Fingern übersehen. Sie wird anstelle der Standardtastatur angeschlossen, soll ein besseres Schreibgefühl bieten (Entschuldigung, ich vertrete eben immer noch die Minderheitsmeinung, daß die Mega-Tastatur ganz und gar nicht abartig schlecht ist) und ist voll "kompabeatle". Für 589 Öhrchen können Sie Ihren Künstlerfingern ein völlig neues Leben bieten. Sollten Sie sich übrigens selbige beim ewigen TOS-EPROM-Wechseln bereits blutig geschunden haben, bietet sich für Sie die "NEXTrombank" für den Mega ST an, auf der zwei Betriebssysteme und zwei ROM-Module Platz finden: umschalten statt aushebeln. 89 Mark kostet dieser Fingernagel/Schraubenzieher/Platinenschoner.

CSH, die Supra-Generalvertretung in Deutschland, vermeldet, daß die Supra-Einbauplatten für den Mega ST nun unter anderem mit Conner-Laufwerken bestückt werden, die sehr leise sind und gerade mal 5.1 Wätter ziehen - selbst für schmalbrüstige ATARI-Netzteile kein Problem. Wir werden uns mit diesen Einbauplatten noch zu beschäftigen haben. Neben dem SCSI-Interface von SUPRA und einem Teil der breiten SUPRA-Plattenpalette präsentierte man auch Hayes-kompatible SUPRA-Modems (2400 Baud).

Bei **HARD & SOFT A. Herberg** war am Messe-Sonntag von einem vom Trubel etwas angeschlagenen Firmenvertreter zu hören, daß man an einem neuen, eigenen Plattentreiber arbeite, der auch zum neuen ATARI-HDX kompatibel sein soll. Im Moment wird allerdings noch die ICD-Software ausgeliefert.

Gleich um die Ecke war die Firma **FSE** aus Kaiserslautern zu finden. Mit einer eigenen Festplattenserie (32 bis 85 MB, 1298 bis 2798 DM,

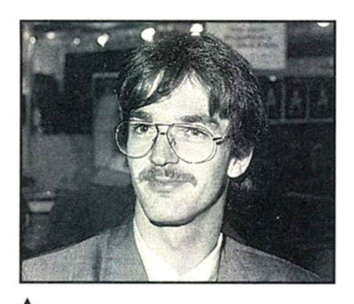
Stimmen zum TT

Die Hauptattraktion der ATARI-Messe '89 war unbestreitbar die Präsentation des ATARI TT, der 32-Bit-Maschine auf Basis des Motorola-Prozessors 68030. Wir fragten einige, die seit langer Zeit auf dem ST-Markt aktiv sind, nach ihrer Meinung über ATARIs neues Flaggschiff.



▲

T. Maier (TommySoftware): "Eine interessante Maschine, die ihre Stellung im ST- und Mittelpreismarkt hat und von der zu hoffen ist, daß sie ihren Platz erkämpfen wird."



▲

U. Eickmann (Eickmann Computer): "Der TT scheint eine interessante Maschine zu sein, die vermutlich ihren Platz im Markt finden wird. Es ist abzuwarten, wie kompatibel sie zum ST ist, damit die Software auch weiterhin läuft. Im Unix-Markt hat der TT sicher in kleineren Bereichen Zukunft."



▲

H.H. Huth (Application Systems): "Ich gebe meine Meinung dazu ab, wenn der TT ein kaufbares Gerät ist, vorher nicht. Aber natürlich ist es ein äußerst wichtiges Gerät für ATARI. Um zu zeigen, daß Innovation da ist, daß die Leute etwas tun. Genauso, wie - von meiner Markteinschätzung her - Mercedes einen SL baut und mehr 190er verkauft, werden sicher bei ATARI durch den TT mehr 1040er verkauft werden."



▲

P. Noone (C.A.S.H.): "Ich habe mich mit dem TT wenig beschäftigt, weil das System für unsere Zwecke und für unsere Firma relativ uninteressant ist. Ich sehe den TT - anfangs - in der Hauptsache im Universitätsbereich. Für Fakturierungen oder Buchhaltungen in dem Stil wie wir sie fertigen, ist das Gerät zu groß. Da müßten wir uns dann Unix-Systemen zuwenden, Mehrplatzfähigkeit usw. In die Richtung haben wir momentan keine Bestrebungen."



▲

J. Springer (Bavaria-Soft): "Tolles Design. Was er leistet, wird man sehen."

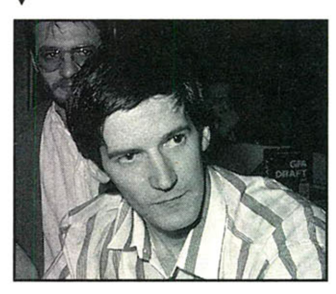
N. Eder (Bavaria-Soft): "Der TT ist eine innovative Maschine. Der 68030 wird mit Sicherheit der Prozessor sein, der in der Zukunft in dem Bereich üblich sein wird, in dem jetzt der 68000 vorherrscht. Für unsere Produkte freuen wir uns darauf, denn dadurch wird auch von der Verarbeitungsgeschwindigkeit her einiges geboten."

▼

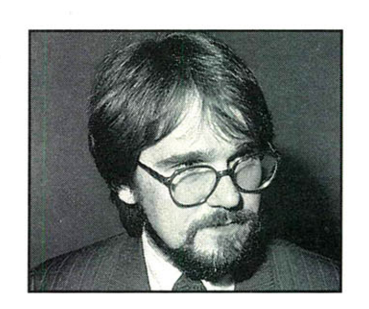


F. Ostrowski (GFA): "Ich habe ihn bisher nur ganz kurz, ganz flüchtig gesehen."

▼



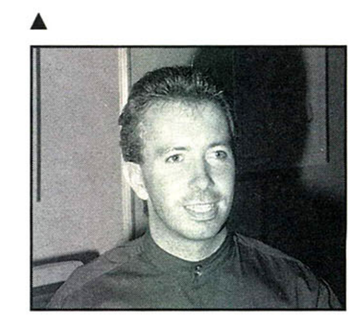
R. Mollenhauer (Adi): "Die TTs werden bei den Benutzern, die jetzt STs haben, später genauso ankommen, ähnlich dem Fahrschuleffekt. Sie werden den TT als nächsten Rechner avisieren. Es gibt also sehr viele, die wahrscheinlich jetzt schon denken: Es gibt einen TT, dann fang ich schon mit einem ST an. Der TT hat aber



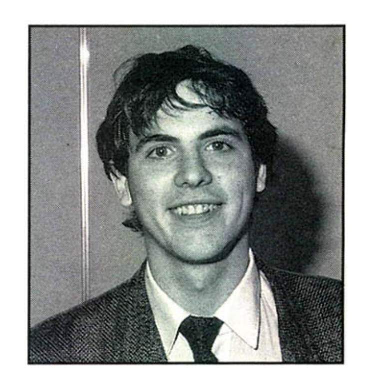
heute ja noch keine Aktualität, weil man ihn nicht im Laden kaufen kann. Der TT ist unseres Erachtens eine wichtige Weiterentwicklung, weil die ST-Kunden dadurch den Eindruck haben, daß es von ATARI etwas in der professionellen Linie gibt, einen Nachfolger."

Dr. Roggenkamp (Rhothron): "Ich halte den TT für eine Maschine, die sicherlich die Produktlinie nach oben hin erweitert und den ATARI für Anwendungen brauchbar macht, die bisher mit dem Mega ST in der unveränderten Form nicht machbar sind."

▲



C. Kraus (OMIKRON.): "Der ATARI TT wird sicher seinen Weg gehen, wenn er entsprechend von ATARI eingeführt wird. Das Wichtigste ist erst ein-



mal, daß Software auf dem TT entsprechend verfügbar wird. Vorher kann das Gerät sich ganz bestimmt nicht durchsetzen. Es wird jetzt vielleicht von einigen Freaks und von Softwarehäusern gekauft. Aber bevor nicht die Anwendersoftware läuft, wird sich der TT nicht durchsetzen."



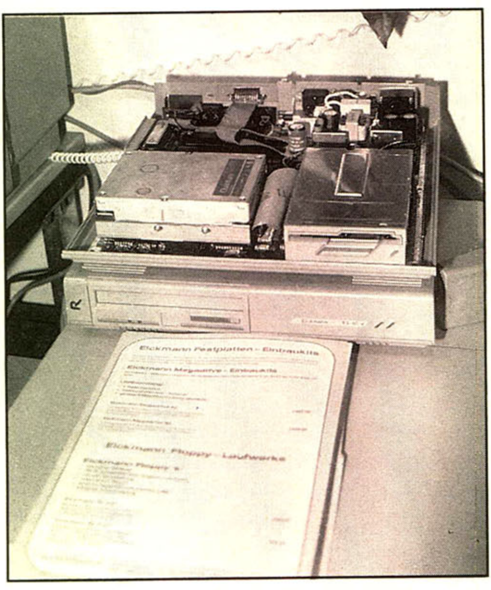
▲

A. Södler (OMIKRON.): "Der TT ist eine sehr gute Maschine. Das Problem ist nur: Es gibt ihn und es gibt ihn nicht."

Autopark, ICD-Treiber) und einer eigenen Wechselplatte zu einem Preis von 1998 Märkern (Medium kostet extra, und zwar 298 DM) will das PC-erfahrene Unternehmen die ATARI-Anwender für sich einnehmen.

VORTEX gab bekannt, daß in allen Platten nur noch Controller eines einzigen Typs ausgeliefert würden. Nicht mehr OMTI, auch nicht Adaptec, nur noch Western Digital 1004. Auch Vortex bot einen Streamer namens SIDEWINDER an, der 150 MB schluckt (3398 DM). Die Vortex-Wechselplatte SIDELOADER (44 MB, 25 ms, die Daten kennen wir zur Genüge) verhält sich wie die neueste Serie der HDPLUS-Platten: Per Tastenkombination kann man die Platte während der Arbeit ausschalten, um die Ohren zu schonen. Optional ist auch ein Zeitintervall einstellbar, nach dem die Platte automatisch ausgeschaltet wird, wenn kein Zugriff mehr erfolgt. Beim nächsten Zugriff wird die Platte automatisch wieder hochgefahren. Hübsch, gell? Im Preis von 2498 DM für den SIDELOADER ist ein Medium und natürlich die Software enthalten, die auch den Medienwechsel unterstützt. Übrigens ist nun auch Vortex mit LANCELOT, einer Ethernet-Karte für den Mega ST, beim Netzwerken dabei.

Jetzt voll im Plattenfieber, blieb auch **EICKMANN Computer** nicht von meinem Besuch verschont. Wie bei anderen Plattenherstellern hatte man hier ein Problem, nämlich, auf der hektischen und lauten Messe zu demonstrieren, wie besonders leise die neuen Festplattenmodelle seien. In diesem Falle die Eickmann-Platten der L-Reihe, die es mit 60 und 120 MB Kapazität gibt (2298 und 3298 Mark). Die Lautstärke wird mit weniger als 40 db(A) angegeben, die Transferrate mit 750 kB/s und der Zugriff mit 24 ms. Alle Plattenpreise sind bei Eickmann übrigens kräftig gepurzelt (die EX110 beispielsweise um 500 DM auf 2998 DM). Auch Eickmann Computer macht sich neuerdings der Mega ST-Völlerei schuldig: Die Megadrive-Einbauplatten mit 40 oder 80 MB sollen 970 kB/s umherschaufeln können, und das bei einer Zugriffszeit von 19 ms. Autopark? Klaro. Man arbeitet übrigens auch an neuer Software mit neuen Tricks, aber das ist alles noch sooo geheim...



Mega ST mit eingebauter Festplatte bei Eickmann.

Mit dem MAC-Disketten schluckenden EXchanger liefert Eickmann Computer einen Beitrag zur prosperierenden Aladin-Welt. Völlig neu ist aber der Eickmann Dispatcher, eine Kombination von Hard- und Software, die - von einem Rechner gesteuert - auf bis zu sechs Monitoren Regieanweisungen und Redetexte ausgibt. Damit eignet sich der Dispatcher ideal für Großveranstaltungen, auf denen an verschiedenen Stellen (Diavorführer, Beleuchter, Redner...) Anweisungen gegeben werden müssen. In Vorbereitung ist auch die direkte Ansteuerung von Lasereffekten vom ST aus. Die Software kostet 498 DM, die Hardware zur Ansteuerung von maximal sechs (Monochrom-) Monitoren ebenfalls 498 DM. Auch geeignete Kabel sind bei Eickmann Computer erhältlich.

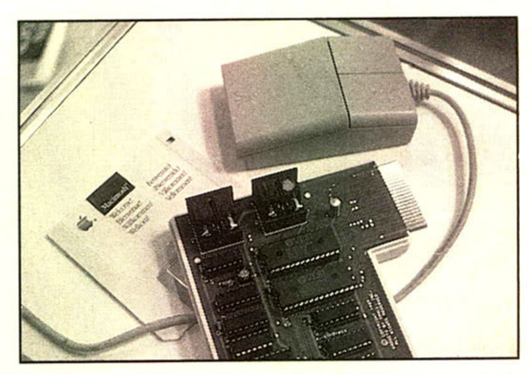
ST emulat

Ein bisher nur sehr schwer zu fassendes Phantomwesen materialisierte auf der ATARI-Messe dieses Jahres im Diesseits: Der Supercharger soll in kleinen Stückzahlen bereits ab Anfang September an Händler, Presse und erste Kunden ausgeliefert werden. Im Preis von 798 DM sind 512 kB RAM und, durchaus unüblich, MS-DOS 4.01 enthalten. Der Supercharger - lötfrei am DMA-Port des STs anzuschließen - bietet einen Sockel für einen Mathe-Coprozessor sowie Erweiterungsmöglichkeiten für mehr internes RAM, das übrigens dereinst über spezielle Software vom ST aus als RAM-Disk nutzbar sein wird. Die ATARI-Maus "belegt" beim Supercharger immer den seriellen Port COM1: des emulierten PCs; daher ist DFÜ-Software auf dem Supercharger nicht lauffähig, man muß dazu schon in den GEM-Teil umschalten. Das immerhin geht sehr einfach: Ein "Hotkey" schaltet direkt zwischen Supercharger- und GEM-Modus um; das Programm im Supercharger wird dabei praktisch eingefroren. Sogar Windows 286 soll auf dem Supercharger laufen, weil der eingebaute V30-Prozessor (ein 8086-Clone von NEC) auch einige 80286-Kommandos beherrscht. Um den am häufigsten gebrauchten Satz von Supercharger-Interessenten zu zitieren: Wurde auch Zeit...

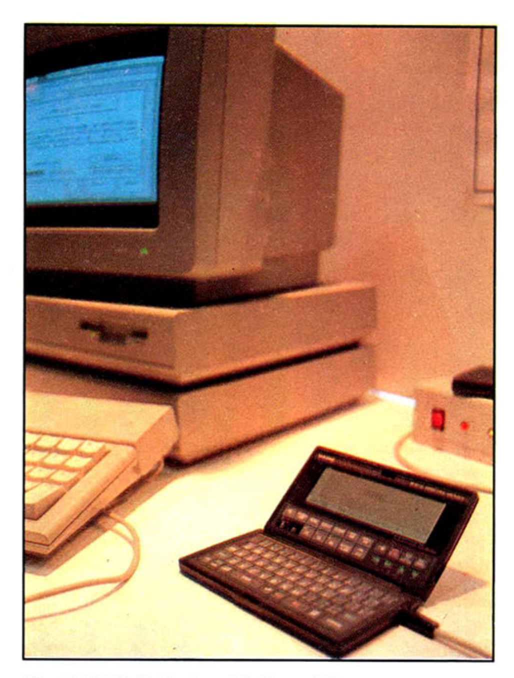
Auf dem Mac-Emulatoren-Sektor war der Spectre GCR bei der Firma **Fearn & Music** zu sehen. Bei dem GCR handelt es sich um die neueste Version des Spectre 128, mit dem sich jetzt auch direkt schnell Mac-Disketten lesen und schreiben lassen. Als besonderer Gag konnte man an einem Messetag direkt via Fax Fragen an den Entwickler Dave Small stellen.

HYPERCACHE, der 16 MHz schnelle und mit 8 kB Cache-Speicher versehene 68000 für den ST, wurde unter allgemeiner Bewunderung am Stand von **proVME** vorgeführt. Eigentlich ist HYPERCACHE ja ein "Abfallprodukt" (Zitat vom Stand) der von proVME entwickelten ST-kompatiblen VME-Bus-Karte, die satte 4161 DM kostet und ebenfalls eine 16 MHz schnelle CPU beherbergt. Intern hat proVME schon eine HYPERCACHE-Version mit einem 68020, die aber nur dann offiziell werden wird, "wenn ATARI ein 68020-kompatibles TOS herausbringt". Eine Steigerung um den Faktor 4 gegenüber dem Normalbetrieb soll mit dieser Karte möglich sein. Übrigens lieferte HYPERCACHE ST bei Benchmark-Tests von proVME ein um nur 20 Prozent schlechteres Ergebnis als ein TT. Bei den Benchmark-Tests handelte es sich, laut proVME, um die bekannten Tests der amerikanischen Zeitschrift BYTE. Nicht überliefert ist, ob beim Test auf dem TT dessen Prozessor-Cache aktiviert war. Einstweilen begnügt man sich bei proVME aber mit einer anderen Neuentwicklung namens HYPERCACHE AMIGA (Gemeinheit!), die zwar nur auf 14.3 MHz getaktet wird (doppelter Systemtakt des AMIGA), aber mit 16 statt 8 kB Cache-Speicher ausgestattet sein wird.

Ebenso in Sachen Emulation und Anpassung an andere Rechner präsentierte auf der ATARI-Messe die schon bekannte Firma **Yellow-Computing** ihr neues Interface zwischen dem ST und der Datenbank-Pocket-Rechner-Serie von CASIO SF 7000, SF-7500 sowie SF-800. Mit der neuen Lösung, die unter dem Namen Transfile SF-700 angeboten wird, ist es möglich, Daten, die auf dieser Pocket-Datenbank gesammelt wurden, auf den ATARI zu übertragen, dort zu sammeln und weiterzubearbeiten oder mit anderen Programmen zu



Der Spectre GCR wurde von Fearn & Music präsentiert.



Der CASIO findet Anschluß am ST.

mischen. Die erweiterten Daten können selbstverständlich zurück in die CASIO-Rechner übertragen werden. Die mitgelieferte Software beinhaltet einen eigenen Editor, der das Bearbeiten von Datensätzen sehr vereinfacht. Die Daten können sowohl im Standard-Casio-Format sowie im Sharp IQ-7000- und im PSION-Organizer-Format abgespeichert werden.

Industrielle Anwendungen...

... und Laboreinsätze verlangen zumeist Bussysteme nach Standards wie IEEE488 oder VME. Erst so ist es möglich, mit Rechnern wie dem ST auf vorhandene Meß- und Steuersysteme zuzugreifen. Zwei Firmen, die in diesem Bereich von jeher sehr aktiv sind, sind IBP aus Hannover und Rhothron aus Homburg.

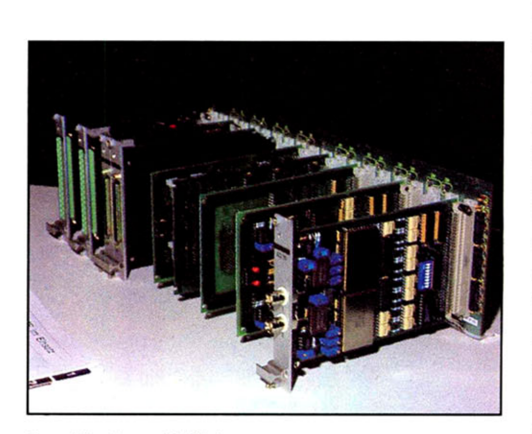


Der IBP 190

Die Hannoveraner, die vor allem durch ihren 190 ST, einem ST auf einer Einschubkarte für 19"-Racks, bekannt wurden, zeigten gleich mehrere Neuigkeiten. Der 190 STE ist die rechenstarke Fortsetzung des 190 ST. Er besitzt nun einen 68020 von Motorola anstelle des MC 68000, der zudem mit 16 MHz getaktet ist. Im Unterschied zur PAK68 arbeitet der 190 STE mit 32 Bit-Speicherzugriffen. In das tragbare VMEbus-System wird fortan ein LC-Display eingebaut werden. Damit ist im Gehäuse mehr Platz für Erweiterungen. Die Anwendungen für dieses System: portables Messen, Steuern, Regeln. Auch im 19"-Rack fallen viele Daten an, und somit stellt **IBP** einen Streamer als Backup-System für Harddisks zur Verfügung.

Bei vielen industriellen Kunden, so war von IBP zu erfahren, konnte man die ST-Systeme bisher nur schwer absetzen, da viele Firmen ausschließlich IBM-kompatible Systeme kaufen. Auch wenn das für den Einzelfall nicht unbedingt sinnvoll ist, erschweren solche firmeninternen Vorschriften das Geschäft. So hat IBP beschlossen, den 190 STX anzubieten. Das ist ein 190 ST, der aber zusätzlich kompatibel zum IBM PC ist. Das System basiert auf dem MS-DOS Hardware-Emulator 'PC-Speed' und wird auch mit der zu diesem gehörigen Software ausgeliefert.

CAD, CAM, CAI, CIM und Verwandte sind als Modewörter übrigens völlig out. Angesagt ist - laut **Rhothron** und ATARI - eher CDI. Zu gut neudeutsch: Computer Designed Instrumentation, ein Begriff, der das Gemeinte allerdings nur nebelhaft umschreibt. Rhothron versteht darunter ein integriertes Meßlabor auf Basis eines ST, der um einen VME-Bus erweitert in einem großen 19"-Gehäuse mit bis zu 20 VME-Bus-Karten sitzt. Für dieses Meßlabor sind bisher folgende Module lieferbar: Speicheroszillator, Logikanalysator, Vielkanalschreiber, Frequenzzähler (bis 100 MHz), Rechteckgenerator, Strom-/Spannungs-/ Leistungsmesser und Temperaturmeßgeräte. Der Clou bei der ganzen Sache ist aber die Rhothron-Software, integriert im rho-softlab. In einem eigenen Desktop können Meßgeräte über Icons angewählt werden; die Ergebnisse von Messungen kommen nicht wie früher per Hand aufs Papier, sondern auf die Platte oder Diskette. Diese Ergebnisse werden mit spezieller Software analysiert und aufbereitet (Regressionsanalyse, Statistik), in Grafiken umgesetzt, mit Versuchsnotizen versehen, per Textprogramm kommentiert und schließlich sogar per DTP-Programm zu einer kompletten Publikation zusammengefügt. Wo früher viele Stationen nötig waren, an denen viele Reibungsverluste auftraten, bleibt heute alles von der Meßwerterfassung bis zur fertigen Aufbereitung in einem einzigen Rechner - unserem guten alten ST eben.



Das Rhothron-Meßlabor

Dazu bietet Rhothron einerseits eigene Software zur Prozeßsteuerung, Meßwerterfassung und Analyse an, andererseits aber auch Standard-Software wie Wordplus, STAD, GFA-Statistik und Calamus zur Aufbereitung. Da Rhothron mit Funktionsbibliotheken zu den VME-Bus-Karten et cetera recht freigiebig umgeht, ist es auch sinnvoll, daß man vom Rhothron-Desktop Programmiersprachen wie GFA- und OMIKRON-BASIC, TURBO-C, ST-PASCAL, Prospero-FORTRAN und Assembler aufrufen kann. Der ST ist bereits jetzt in Wissenschaft und Forschung gut vertreten, und die CDI-Idee sollte das noch fördern. Zumal man bei dieser Lösung nicht nur Zeit und (Konvertierungs-)Streß spart, sondern auch bares Geld.

Mega STs, die Rhothron ausliefert, werden ab sofort serienmäßig mit einer IEEE-488-Schnittstelle ausgeliefert. Diese Schnittstelle entspricht dem HP-IB- oder IEC-625-Bus und ist damit ideal geeignet, um die exotischsten Meßgeräte auch am ST zu betreiben; man spart sich so immerhin einen vergleichsweise teuren HP-Rechner. Ein auch als Programm lauffähiges Accessory sorgt für die Ansteuerung der Meßgeräte; Funktionsbibliotheken in den oben erwähnten Sprachen werden mitgeliefert.

Rhothron kündigte weiterhin eine Hardware-Erweiterung für den Mega ST an, mit der dieser als 16-Kanal-Analogschreiber brillieren kann; den TT wolle man schon bald mit "über 40" Karten und der passenden Software unterstützen. Bei dem Gehäuse hat er auch alle Unterstützung nötig...

Der Stoff, von dem der Rechner lebt

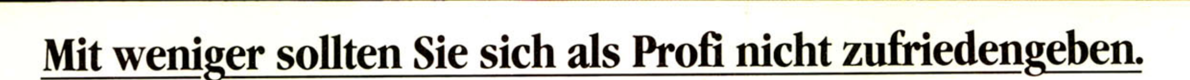
Genug Hardware? Meine ich aber auch. Verbeugen wir uns zunächst gen BASIC-Mekka: **OMIKRON** zeigte MORTIMER, ein multifunktionales Accessory, das ab Oktober für 79 DM zu haben sein soll. Die wichtigsten Funktionen: Texteditor, Drucker-Spooler, Tastaturmakros, Virenwächter (auch gegen Linkviren), flexible RAM-Disk, Snapshot, Bilderanzeiger (Bild im Desktop anklicken, und schon wird es einfach angezeigt), Taschenrechner, Plattenwächter

C&C Computers and Communications

Verfügbar beim Fachhandel.

Pinwriter P6 plus. Der Nachfolger einer Legende.





Kennen Sie die Situation: Ihr Drucker soll ein längeres Dokument ausgeben und blokkiert Ihren PC, weil sein Pufferspeicher nicht ausreicht und er nicht schnell genug druckt.

Oder diese: Sie haben eine tolle Geschäftsgrafik entworfen und finden auf dem Papier nicht das Ergebnis, das Sie erwartet hatten.

Wenn Sie eine dieser Fragen mit Ja beantworten, dann haben Sie nicht den richtigen Drucker. Mit dem NEC Pinwriter P6 plus kann Ihnen so etwas nicht passieren.

Er ist der Nachfolger des legendären Pinwriter P6, mit dem NEC einen neuen Standard für 24-Nadeldrucker setzte.

Der Pinwriter P6 plus übertrifft seinen berühmten Vorgänger noch einmal. Er druckt schnell (bis zu 265 Zeichen pro Sekunde), gestochen scharf (bis zu einer Auflösung von 360 x 360 dpi) und ist großzügig mit Speicher ausgestattet (80 KByte Pufferspeicher, bis zu 50 Seiten).

Weitere Pluspunkte:
- Farboption nachrüstbar
- Papierparkfunktion
- Carbonband
- Schriftartenkarten

Übrigens: Als NEC Pinwriter P7 plus kann er auch DIN A 3 verarbeiten.

Ihr NEC Fachhändler erwartet Sie.

Beachten Sie bitte: Nur der Pinwriter P6/P7 plus mit dem umfangreichen deutschen Handbuch, Druckersoftware und der 12-Monats-Garantie (incl. Druckkopf) ist das Original der NEC Deutschland GmbH.

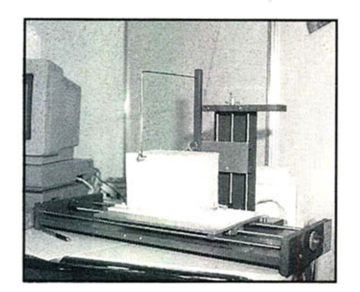
Sag ja zu NEC.

Weitere Informationen erhalten Sie von:
NEC Deutschland GmbH · Klausenburger Straße 4 · 8000 München 80 · Telefon: 0 89/9 30 06-3 45
Fax: 0 89/93 77 76/8 · Telex: 5 218 073 und 5 218 074 necm d



ATARI und die Schüler

Nicht gerade einfach tut sich der ATARI ST im Bereich Schulbildung. Die etwas versteiften Ansichten verschiedener Kultusministerien in verschiedenen Bundesländern sperren sich vor dem Einsatz dieses leistungsstarken Rechners im Schulunterricht. Der Einsatz von Rechnern i.allg. wird nicht bestritten, es herrscht aber die Meinung, es müßte ein Rechner sein, der zumindestens so ähnlich aussehen sollte, wie der mit den drei bläulichen Buchstaben. Lehrer, die mit ATARI ST bessere Erfahrung gemacht haben, z.B. in Rheinland-Pfalz oder im Ruhrgebiet, setzen trotzdem den ST ein, und wenn MS-DOS-Software vorgeschrieben wird, dann benutzen sie einfach einen Emulator, dann verhält sich der ST, wie es von dem Kultusministerium vorgeschrieben wurde.

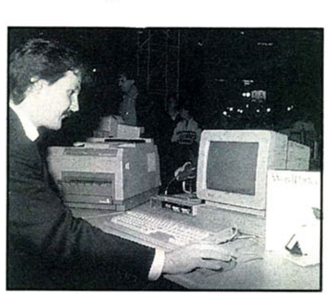


Eine ST-gesteuerte Schneidemaschine

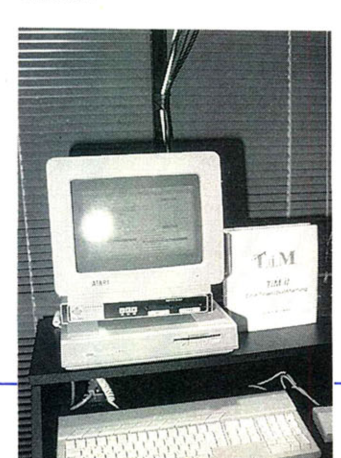
Weiterhin haben die Lehrer einen Selbsthilfeverein gegründet, über den der Austausch von Information über Hard- und Software im Schulungsbereich vereinfacht werden kann. Schulte Borbeg, ein Lehrer aus Wickede an der Ruhr, zeigte auf der Messe in Düsseldorf verschiedene Ein-/Ausgabegeräte, die von dieser Gruppe teilweise entwickelt wurden und in vielen Schulen die Grundaustattung für den Unterricht bilden. Wir werden in einem anderen Heft noch mehr über dieses Thema berichten.

STs im Netz

Mit drei 'Bürolandschaften', die auf dem ATARI-Stand aufgebaut waren, ist es ATARI gelungen, die professionellen Einsatzmöglichkeiten für den ST unter Beweis zu stellen. Zu jeder dieser Landschaften gehörten mehrere Tische, auf denen an jeweils vier miteinander vernetzten Rechnern gearbeitet wurde. Zum Einsatz kamen die Netzwerke von GTI (eLAN), Biodata (Bionet) und Rhothron (Rhonet). Leider war die optische Aufmachung dieser, eigentlich als Schwerpunkt der Messe konzipierten Büros nicht sehr gelungen. So wurde es den meisten Besuchern gar nicht klar, daß die Rechner vernetzt waren und was dort eigentlich demonstriert wurde. Außer den Tischen selbst erinnerte auch nichts an ein Büro.



Eine dieser Bürolösungen zeigte das Zusammenspiel von 'TechnoCAD' zur Zeichnungserstellung, 'Calamus' zum Erstellen der nötigen technischen Dokumentation unter Einbeziehung der Zeichnungen und 'Depot' zur Angebotserstellung unter Zuhilfenahme der Stücklisten, die zur Zeichnung gehören. Dieses integrierte System ist keine Fiktion; es ist für einen deutschen Lautsprecherhersteller entwickelt worden.





Netzwerk und damit Büro Nummer zwei zeigte 'MegaPaint II', 'Calamus' und das neue 'Word Perfect' beim Zugriff auf einen Laserdrucker und einen gemeinsamen Datenbestand. Dieser wurde von einem ATARI PC5 (einem 386er AT), der sich problemlos in das Bionet einfügt, verwaltet.

Die dritte Tischgruppe schließlich wurde vollständig von Adi bestückt. Über das Netzwerk eLAN von GTI kommunizierten die neuen Versionen von Adimens (3.0) und das ebenfalls modifizierte AdiTalk miteinander.

(meldet Plattenzugriff, wenn ein angeschlossener Laser ausgeschaltet ist), Diskfunktionen, Plattenparken, Bildschirmlupe. Hapüh, erst mal wieder durchatmen.

Für OMIKRON.BASIC (an diesen Punkt werde ich mich nie gewöhnen) gibt es jetzt eine Finanzmathematik-Library (400 Funktionen, 129 DM) und die SQL-Library. SQL ist eine standardisierte Datenbankabfragesprache von hoher Komplexität; für 298 DM sind Sie auch in OMIKRON.BASIC dabei. Der OMIKRON-Assembler (wir reden von der Grundversion) ist bei der Versionsnummer 1.86 angelangt: Der Editor erfreut sich neuer Optionen (Suchen und Ersetzen von Labels mit Wildcards), der Linker beherrscht das DRI-Format, der Assembler hat eine Menüzeile, und das Handbuch ist überarbeitet worden.

Herr Sender von **COMPUTERWARE** freute sich, die Version 2.05 von Neodesk präsentieren zu können; es hat nun eine Schnittstelle zu Accessories, die in späteren Produkten ausgenutzt werden soll. Außerdem sind die Icons redefinierbar geworden. Um das einfacher zu machen, gab es bei COMPUTERWARE das "Icon Construction Set", kein Programm etwa (auch der Autor dieser Zeilen fiel diesem Irrtum zum Opfer), sondern ein Entwurfsblock für Icons, in dem man nach Herzenslust Bilderchen malen konnte. Ideal für die Kleinkindphasen nach dem fünften oder sechsten Messebier am Abend...

Wie begehrt eine komfortable Dateiauswahlbox ist, belegt die Beliebtheit der ST-Computer-PD-Diskette, auf der die Box des Herrn Patzel zu finden ist. Dieses Zeichen der Zeit hat man auch bei COMPUTERWARE erkannt und bietet den "Universal Item Selector" an, eine große Dateiauswahlbox mit vielen Optionen (Formatieren, Kopieren, Verschieben, Umbenennen, Löschen, Ordner-Anlegen, Schreibschutz für Dateien, Dateisuche...), als Programm für den Autoordner. Dieses Programm wird zusammen mit HERMES, einem IMAGIC-Produkt, geliefert, das mir einen alten Wunsch erfüllte: Endlich ein Bildschirmschoner mit Mac-ähnlichem Feuerwerk!

HERMES beherbergt noch weitere Bonbons wie Goodblit (behebt Busfehler bei einigen Blitter-Baureihen), Mausbeschleuniger, Paßwortabfrage beim Booten, Pausefunktion (Rechner wird blockiert, bis wieder das Paßwort eingegeben wird), Speicherlöscher, 50/60-Hz-Umschaltung und Tastatur-Reset. HERMES und der Item Selector kosten zusammen 69 DM.

Noch ein Textprogramm! Ab November will COMPUTERWARE das Programm TYPOGRAF gegen jeweils 169 Mäuse eintauschen (nein, nicht die grauen mit dem Kabel dran). Diese neue Textverarbeitung kennt einen schnellen Editormodus; ein eigenes Desktop mit Papierkorb, Laufwerken und eigener Dateiauswahlbox macht die Bedienung recht einfach. Maximal 20 Texte können parallel bearbeitet werden. TYPOGRAF kann im Text rechnen, alle Tasten frei belegen, im Hintergrund drukken, Breit- und Großschrift auf dem Bildschirm darstellen (natürlich auch kursiv und fett) sowie Stichwort- und Inhaltsverzeichnisse erstellen. Die Texte werden wahlweise auch mit Zeilennummern angezeigt. Serienbriefe sind ebenfalls kein Problem. Man wird sehen - im November. Die ersten Demos auf der Messe waren recht vielversprechend.



Die tausend Hände von gdat

gdat zeigte sich ebenfalls umtriebig. GD-Fibu, die "Euro-Finanzbuchhaltung", bucht in beliebigen Fremdwährungen nach EG-Richtlinien: Gewinn- und Verlustrechnung, Bilanz, offene Posten, Kostenstellen und Skontoberechnung sind hier in einem leicht bedienbaren und konfigurierbaren Programm zusammengefaßt.

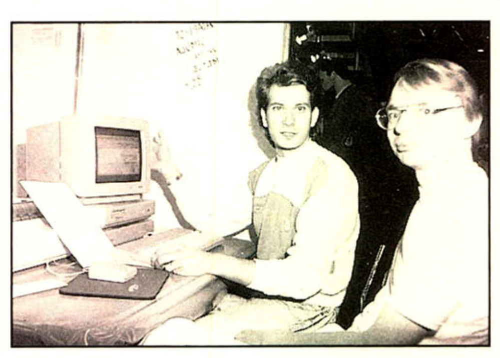
Für MIRAGE gibt es nun ein multiuser- und multitasking-fähiges ANSI-C; der Compiler erzeugt wahlweise auch 68020-Code. APL/68000 gibt es nun auch für 68020 samt Coprozessor (PAK-Karte), wodurch APL für Anwendungen interessant wird, in denen allerheftigst Zahlen durch den Prozessor geschleift werden. Das APL/68000 für den Mac bietet gdat übrigens nun auch auf Disketten im ALADIN-Format an. Das MAC-APL ist multitaskingfähig.

Besonders stolz ist man bei gdat auf GD-SYS, ein Entwicklungssystem für APL-Programmierer. GEM quält Programmierer mit vielen lästigen Arbeiten wie Bildschirm-Redraw und Ereignisbehandlung. Das alles soll mit GD-SYS um einiges einfacher werden. GEM-Fenster können in Felder mit individuellen Scrollbalken unterteilt werden, die schnellen Scroll-Routinen stoßen sich auch nicht mit Großbildschirm. Mitgeliefert werden einige fertige Tools, etwa ein Formatierprogramm oder eine Fileselectbox mit assoziativer Suche. Um gdat zu zitieren: Allein GD-SYS sei schon "ein guter Grund, auf APL umzusteigen".

Auf dem ATARI ST ist jetzt auch eine Portierung des bekannten Simula-Systems erhältlich. Es handelt sich dabei um eine objektorientierte Sprache, die mit Compiler, Linker, symbolischem Debugger, Schnittstellen zu C, Fortran und Pascal und ausführlicher Dokumentation geliefert wird.

Borland will mir offensichtlich einen meiner Herzenswünsche erfüllen: Anfang nächsten Jahres soll es Turbo-C 2.0 samt dem brandneuen Turbo-Debugger geben, der endlich auch im Source-Code nach Wanzen fahndet. TC2.0 wird eine Option bekommen, um dem fertigen Programm zusätzliche Informationen für das Source-Level-Debugging mitzuliefern. Weitere Verbesserungen für den Compiler stehen noch nicht fest.

Der Debugger emuliert das gesamte GEM (und ist dabei schneller als das Original), weil GEM selbst nicht reentrant ist. So wird auch das Debuggen von GEM-Anwendungen - auch auf Großbildschirm - möglich: Daß man im Turbo-Debugger plötzlich beliebig viele Fenster öffnen kann, verwundert dann auch nicht weiter. Nur Accessories können aus dem Turbo-Debugger nicht aufgerufen werden. Der Turbo-Debugger soll - wie die integrierte TC-Oberfläche - kontextsensitiv Hilfe leisten können: Man klickt also einen Begriff im Programmtext an ("printf") und bekommt sofort alle Informationen aus dem Handbuch darüber. Alle 680xx-Prozessoren sollen unterstützt werden, also auch der 68030 des TTs. Variablen, Felder und Zeiger können während des Programmlaufs beobachtet werden. Dabei kann ein Programm komplett durchlaufen werden und an Breakpoints halten, für die komplexe Bedingungen angegeben werden können (C-Ausdrükke, Veränderung von Variablen und Speicherinhalten), oder aber in Zeitlupe per "Animate" begutachtet werden.



Die Simula-Entwickler am ST

Aus gegebenem Anlaß besuchte ich - obwohl eigentlich Spielehasser - auch den Stand von **THALION** auf dem Ariola-Komplex in Halle 1. Umgeben von Horden von Schulkindern, konnte ich dort Neuentwicklungen begaffen wie "Dragonflight", ein komplexes D&D-Spiel mit Animationssequenzen, "Seven Gates of Jambala", ein Jump'n'Run-Spiel mit 7 Höhlen und Querlevels, "No 2nd Prize" (Arbeitstitel), ein futuristisches Rennspiel in sagenhaft schneller 3D-Vektorgrafik, bei dem mehrere Rechner über MIDI und Centronics gekoppelt werden können, zuletzt auch "Chambers of Shaolin", ein Kung-Fu-Kampfspiel. Alle Spiele glänzen durch guten Sound und saubere Grafik.

ATARI ST - der Grafikrechner

Auf der Düsseldorfer Messe wurden eine ganze Reihe von Entwicklungen im Hard- und Software-Bereich gezeigt, die die Bedeutung des STs für grafische Anwendungen unterstreichen. Eines der Highlights war die Verwendung des Epson Farbscanners GT-4000. Dieses Gerät ist ein absoluter Preisbrecher im Scanner-Markt. Für rund 5000 DM bietet er eine Auflösung von 400 DPI bei einer Farbstufung von 24 Bit pro Pixel, also jeweils einem Byte pro Grundfarbe. Nun ist Farbe ja nicht gerade die Stärke des STs und so mußten sich die Entwickler einiges einfallen lassen, um die gescannten Farbbilder auch nur in annähernd vernünftiger Qualität ausgeben zu können. Mit den 256 Farben und der geringen Auflösung, die der ST in seiner niedrigen Einstellung bietet, ist dies nämlich nicht möglich.

In verschiedenen Systemen wurde der Scanner eingebunden, davon ist allerdings der der Firma **TmS** noch nicht marktreif. Hier hat man zunächst das Programm 'TmS Scandesign' auf Farbbetrieb abgestimmt und ist nun in der Lage, die Grafiken mittels einer Grafikkarte von Matrix oder der MGE von MAXON auf entsprechenden Multisync-Monitoren auszugeben. Die dabei erzielte Qualität ist sehenswert. Nun müssen die Bilder auch zu Papier gebracht werden, dazu dient ein Tintenstrahldrucker von Canon. Der Canon Full Color Printer FP-510 ist in der Lage, mit ebenfalls 24 Bit pro Pixel und einer Auflösung von 160 DPI zu drucken. Noch nicht entschieden ist bei TmS, für welche Grafikkarte man sich letztendlich entscheiden wird, doch muß bis zur Serienreife des Komplettsystems noch kräftig an der Software gearbeitet werden. Diese wird umfangreiche Bearbeitungs-



Der Canon Tintenstrahldrucker druckt farbig.

funktionen für die Farbbilder enthalten. Angesprochene Käuferschicht dieses Systems, das komplett mehr als 20000 DM kosten wird, sind Einsteiger im Grafikmarkt, denen andere Lösungen, die es vor allem für den Apple Macintosh gibt, zu teuer sind.

TmS zeigte in Düsseldorf auch die Version 2.5 des Vektorisierungsprogramms 'TmS Vektor'. Das Update ist erheblich schneller und funktionsreicher als die bisherige Version. Richtig neu aber ist 'TmS Graphics', ein vektororientiertes Zeichenprogramm. Dem Grafiker wird es dienen, schnell Entwürfe herstellen zu können, dabei zu verzerren und zu vergrößern, ohne die Verluste pixelorientierter Programme in Kauf nehmen zu müssen.



Die TmS zeigte massiv Grafikanwendungen.

Eine andere Lösung in diesem Bereich, in dem sowohl der Epson-Farbscanner wie der Canon FP-510 Verwendung finden, kommt von **Marvin**. Die Firma aus der Schweiz hat ein Videofarbsystem entwickelt, dessen Herz eine digitale Videokarte für den ST ist. 'Chili'- so der Name des Systems - ist in der Lage, Videoeffekte zu erzielen, wie sie bei der Produktion von Werbe- und Präsentationsvideos gebraucht werden. Die Software bietet dabei zahlreiche Möglichkeiten, Video- und gescannte Bilder miteinander zu verbinden. Gespeichert werden die riesigen Datenmengen, die dabei auftreten, auf einer löschbaren optischen Platte. Auch das ist ein Novum am ST. Alles in allem ist 'Chili' ein Low-Cost Produkt im Bereich professioneller Genlocks und zielt auf semiprofessionelle Videostudios.

Für mehr Professionalität im Grafik- und CAD-Bereich hat wesentlich die Einführung des 19"-Monitors M110 durch die Firma **Matrix** geführt. Seitdem sind Anwendungen in diesen Bereichen möglich geworden, die bislang Rechnern wie IBM ATs und Apple Macintoshs vorbehalten gewesen waren. Seit ATARI mit dem SM 194 selbst einen 19-Zöller anbietet, herrscht Konkurrenz auf dem Markt der großen Röhren. Matrix zeigte in Düsseldorf den neuen M110L, einen Visa-Monitor, der an die Matrix-Grafikkarte angepaßt wurde und zum Preis von 4200 DM angeboten wird.

Gleichzeitig ist der Preis des 'alten' M110 auf 4600 DM gesenkt worden. Neu im Programm ist ebenfalls der M120, ein 21-zölliger Multisync von Eizo, der in der Lage ist, 256 Graustufen anzuzeigen. Die Grafikkarte C128, die in der Lage wäre, diese Graustufen zu produzieren, gibt es allerdings noch nicht, der M120 wird vorerst mit der bisherigen, rein monochromen Karte ausgeliefert. Die neue soll erst gegen Jahresende serienreif sein. Neben einem Prototyp gab es aber auch schon die technischen Daten zu sehen: 256 Graustufen bei einer Auflösung von 1400x1024 Pixel, 16 Farben mit 1664x1232 Punkten (ideal für CAD) oder 16 Millionen Farben mit 850x512 Bildpunkten für Echtfarbdarstellung sind die Eckdaten.

Die Firma **3K**, die immer noch so heißt, obwohl das mittlere "K" nicht mehr dabei ist, bot den genannten EPSON-Scanner GT4000 zum Messepreis von 4250 DM feil. Einziges Problem: Eine Steuerleitung, die am ST-Centronics-Port unbelegt ist, muß nachgerüstet werden. Dazu bietet 3K einen Umrüstservice an, für den der Kunde nur die Portokosten für das Hin- und Herschicken des Rechners übernimmt; Die Garantie bleibt unberührt. Nach der Messe soll der Scanner 4500 DM kosten.

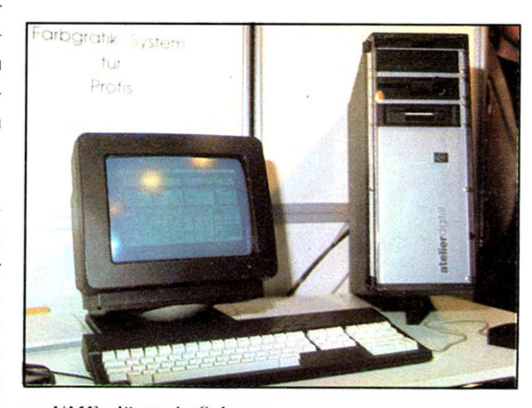
RETOUCHE von 3K ist mittlerweile für seine Künste bekannt (siehe Test in ST-Computer 6/89). Nun wird es mittels eines trickreichen Accessories namens TURBO256.ACC auf Bilder des Digitizers TURBODIZER losgelassen, um bis zu 256 Graustufen zu erzeugen. Ergebnis: Effektiv eine wesentlich höhere Auflösung - man überzeugte mich mit auf Laserbelichtern ausgegebenen Bildern, in denen der Fortschritt deutlich wurde.

Der SCSI-Controller von 3K erfüllt nicht nur die volle SCSI-Norm (alle Kommando-gruppen, nicht nur Gruppe 0) und ist damit zum Anschluß beliebiger SCSI-Medien geeignet, sondern macht mit der passenden Software auch eine einfache Vernetzung möglich. Sieben STs mit 3K-SCSI-Controller und ein Massenspeicher können an diesem Netz hängen (Buslänge maximal sechs Meter). Die SCSI-Software verwaltet bis zu 8 Gigabyte in maximal 512 Partitionen, kontrolliert den Medienwechsel und erlaubt ein einfaches Partition-Locking (greift ein Anwender schreibend auf eine Partition zu, darf ein anderer nur noch lesen). Die Programmierschnittstellen sind offengelegt. Mit 399 DM ist man dabei.

"Every time you try to operate one of these weird black controls that are labelled in black on a black background, a little black light lights up black to let you know you've done it."

Zaphod Beeblebrox etwas entgeistert über den Bedienungskomfort in einem Raumschiff, entnommen dem "Restaurant at the End of the Universe" von Douglas Adams.

Ein ähnlich aufsehenerregendes optisches Erlebnis erwartete den Besucher am proVME-Stand: Ein schwarz gespritzter SM124 samt schwarzer Mega-Tastatur, schwarzem Grafiktablett und schwarzem Farbmonitor zog alle Blicke auf sich. Das edelschwarze Design dürfte die Zielgruppe für dieses System einnehmen: Die Grafiker. Das Ganze nennt sich "atelierdigital" und basiert auf einer ST-kompatiblen VME-Bus-Karte, die mit einer VME-Bus-Grafikkarte in einem Towergehäuse untergebracht ist. Die Grafikkarte produziert auf dem Farbmonitor bei einer Auflösung von 800x600 Punkten 256 Farben aus einer Palette von 262144 Farben, hat einen eigenen Videospeicher von 1 MB und einen Grafikprozessor. Die Software ist für zweidimensionale Vektorgrafik ausgelegt (die interne Auflösung beträgt 20000 Linien), soll sehr einfach zu bedienen sein und unterstützt einige Farbeffekte. Zwei lizensierte Vektorschriften werden für Beschriftungen mitgeliefert. Das



proVME glänzte in Schwarz.


Grenzenlos –
Erlebnisstark
ATARI ST Computer –
da steckt Wahnsinns-Power drin
ATARI
1040ST
Das sind Computer der Spitzenklasse.
Super stark – dabei echt schnell.
Ob spannende Action, Animation, Textverarbeitung, Kalkulation, Grafik, Programmieren oder Musik. Alles geht! Mit Superfarben. Oder, so wie's Profis mögen, Schwarz auf Weiß. Mit dem hochauflösenden ATARI Monitor SM 124.
ATARI ST Computer – Höchstleistung auf allen Gebieten. 2 x „Computer des Jahres".
Superprogramme Signum, Calamus, STAD, Lavadraw, PGRAPH, Imagic, Creator, CV Base, Twenty Four und viele andere gibts nur für ATARI ST Computer.
ATARI
. . . wir machen Spitzentechnologie preiswert.

MAXON und die ATARI-Messe

Auch die MAXON Computer GmbH war mit einem Stand auf der Düsseldorfer Messe vertreten. Zu sehen war die neue Grafikkarte MGE (MAXON Graphic Expansion), die eine maximale Auflösung von 1280x1024 Bildpunkten hat und max. 256 Farben aus einer Palette von mehr als 256000 Farben darstellen kann. Aber nicht nur für den reinen Grafikbereich, sondern auch für DTP auf Großbildschirmen läßt sich die Karte optimal einsetzen. Für einen Preis von DM 1798,- bietet die MGE ein sehr gutes Preis-/Leistungsverhältnis (siehe auch Bericht ST 9/89). Neu war auch der MGP-GAL-Prommer, der ab



Gedränge am MAXON-Stand

dieser Ausgabe als Projekt in drei Teilen vorgestellt wird.

Bei Harlekin, dem neuesten Softwareprodukt MAXONs, handelt es sich um ein Accessory, daß 15 Programme beinhaltet: So ist ein schneller Editor, ein Terminkalender, eine RAM-Disk, ein Disk- und ein Speichermonitor, ein Disketten-Utility zum Kopieren und Formatieren u.v.m. enthalten. Für Leute, die es gern kompakt mögen, bietet Harlekin bei einem Preis von DM 129,- für 15 Programme sicherlich das Ideale. Ferner war die neue SoftArt-Reihe zu sehen: Hier konnte man u.a. den MiniEd, einen Editor als Accessory, das Diskettenverwaltungsprogramm Diskstar und das Multifunktionsprogramm Merkur sehen. Doch auch wem die normale Softwarewelt des STs zu teuer war, der konnte sich regelrecht in die PD-Software der ST-Computer wühlen, da es eine kostenlose Messesonderausgabe der PD-NEWS gab, die neben interessanten Berichten über PD-Software die komplette Liste der ST-Computer-PD-Sammlung enthält.


L.Tramiel begutachtet den neuen MGP-GAL-Prommer.

Ergebnis der Zeichenarbeit kann schließlich für Dia-Belichtungsanstalten verwendet werden. Einschließlich Schulung, Installation und Hotline soll der Spaß 35000 Mark kosten, gegenüber vergleichbaren PC-Lösungen ein attraktiver Preis, wenn man bedenkt, daß das System mit 4 MB Speicher, vier seriellen Ports, zwei Diskettenlaufwerken (3.5" und 5.25") und einer 85 MB-Platte ausgerüstet ist.

Weiter in der Grafikecke: **IMAGIC**, bekannt durch ihre Grafikprogramme, die von Application Systems Heidelberg vertrieben werden, war mit einem eigenen Stand präsent. Das GENLOCK VM410 mit sieben Eingängen (PAL/SECAM, RGB, Computerbild) und einem Ausgang (RGB+Sync) erzeugte hier dank einer installierten Videokamera allerlei Mischmasch von Besucherbildern und Computergrafiken. Das GENLOCK von IMAGIC kann zwei Videobilder mit dem Computersignal mischen. Die Genlock-Farbe ist über Drehregler frei wählbar und daher nicht notwendigerweise immer schwarz. Für 4498.- DM kann man sich die Genlock-Karte in den Mega ST einbauen lassen.

IMAGIC bietet aber auch ein Komplettsystem im 19"-Gehäuse an, in dem ein Mega ST und das GENLOCK-Gerät stecken. Dazu wird allerhand Software geliefert, darunter auch ein Grafikprogramm, das auf dem ST 16 aus 4096 (!) statt 512 Farben darstellt - allerdings nur per Interlace-Verfahren. Dieses Grafikprogramm soll demnächst auch bei Application Systems Heidelberg erscheinen. Bleibt noch zu erwähnen, daß das erwähnte Komplettsystem mit maximal 255 anderen Rechnern koppelbar ist, und daß IMAGIC auch einen Videoprinter präsentierte, ein Gerät, das praktisch Video-Sofortbilder schießt. Bei einer Auflösung von 640x580 Punkten und 64 Intensitätsstufen für jede Grundfarbe waren die resultierenden Bilder, die dieser 3-Farben-Thermodrucker (was anderes ist es nämlich nicht) ausspuckte, recht ansehnlich.

Ein weiteres Grafik-Utility zu 'Signum!' gibt's von **Andreas Pirner** aus Berlin. Der Autor von 'Headline' und 'Fontmaker' präsentiert das Programm 'Convert'. Mit dieser Software ist es möglich, Pixelgrafiken aller nur erdenklichen Formate - sei das vom ST, vom Macintosh oder vom PC - einzulesen und so aufzubereiten, daß sie direkt in Signum!-Dokumente eingebunden werden können. Damit erweitern sich die Fähigkeiten von 'Signum!' als kombiniertes Text- und Grafikprogramm wiederum.

Noch ein Malprogramm: Aus Holland kommt die Firma **Com-Media**, die eine Software zum Aufbereiten großformatiger Pixelgrafiken, wie sie beim Scannen vorliegen, zeigte. 'Image' ist in der Lage, je nach Speicherausbau Formate bis zu 6000x4000 Pixel zu bearbeiten.

Ganz ähnliche Aufgaben erledigt das Programm 'Touch Up' von **Jörg Wilhelm**. Für 398 DM bekommt der Kunde ein Werkzeug, das in erster Linie zur Nachbearbeitung gescannter Pixelgrafiken dient.

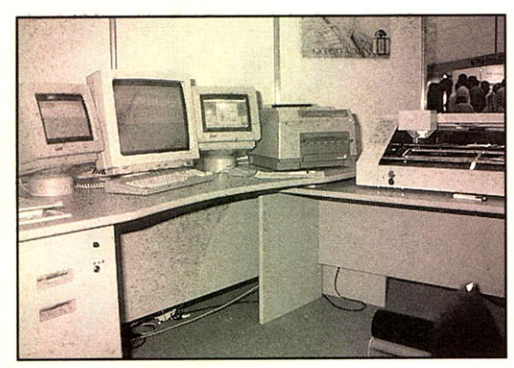


Touch Up - Das Vektorgrafikprogramm von Wilhelm

'MegaPaint II' ist neben 'GraphStar' und 'SoundMerlin' die Neuigkeit bei **TommySoftware** aus Berlin. Die neue Version von 'MegaPaint' enthält wiederum einen ganzen Schwung Funktionen, die weit über die eines herkömmlichen Malprogramms hinausgehen. Dazu gehören Vektorschrift, Bemaßungsfunktionen, Lot fällen, Bezierkurven etc. Erhältlich ist ebenfalls eine Erweiterung für den Großbildschirm und eine Netzwerkversion. 'Graphstar' ist ein Malprogramm, das als Accessory in allen GEM-Programmen zur Verfügung steht. Mit 'SoundMerlin' stellt TommySoftware eine Sample-Software zur Verfügung, die umfangreiche Sample-Abspiel- und Edierfunktionen besitzt. Das Programm kostet 299 DM und ist kompatibel zu den Soundsamplern Prosound, dem von **gdata** und dem Sampler aus eigenem Hause.

Die Firma **VIP-Design** aus Belgien stellte ein Komplettsystem für Grafiker vor, die aus Folien Werbelogos und Schriftzüge fertigen wollen. Der Arbeitsplatz, den VIP-Design liefert, besteht aus zwei Mega STs, einem 19"-Monitor, einem 300 DPI-Scanner und einem Schneide-Plotter nebst eigener Software zum Scannen und Vektorisieren. Der Grafiker hat damit die Möglich-

keit, binnen kürzester Zeit aus einer Vorlage geschnittene Folien für Leuchtwerbung, Messebau oder Fahrzeugbeschriftung herzustellen. Obwohl der Gesamtpreis von weit über 50000 DM zunächst schreckt, ist er doch auf diesem Gebiet - verglichen mit der Leistung und der Geschwindigkeit, die die ATARIs bieten - günstig. Aufgrund der großen Nachfrage planen die Belgier, den zentralen Teil, nämlich das Scan- und Vektorisierungsprogramm 'VIP-1' in einer abgemagerten Version auch einzeln zu vermarkten. Der dabei angestrebte Preis beträgt 380 DM.



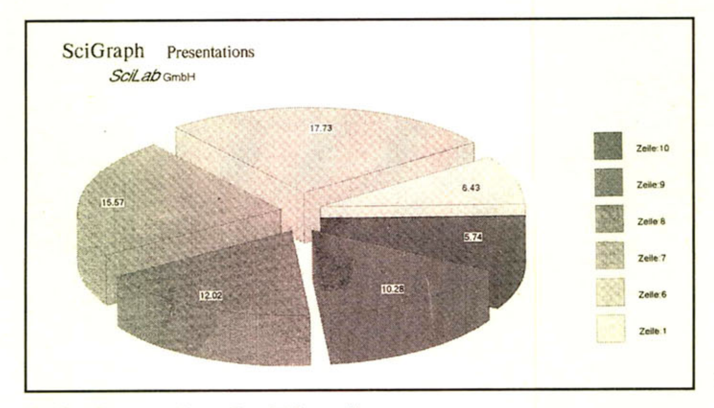
Das Schneidesystem von VIP-Design

Shift stellte einen Zusatz zum pixel- und vektororientierten Zeichenprogramm 'Arabesque' vor. Das Accessory ist in der Lage, Pixelgrafiken, die Teile einer Arabesque-Zeichnung sind, zu vektorisieren. Das ist möglich, da eine Schnittstelle zwischen 'Arabesque' und Accessories besteht, über die auf den Arbeitsbereich des Zeichenprogramms zugegriffen werden kann.

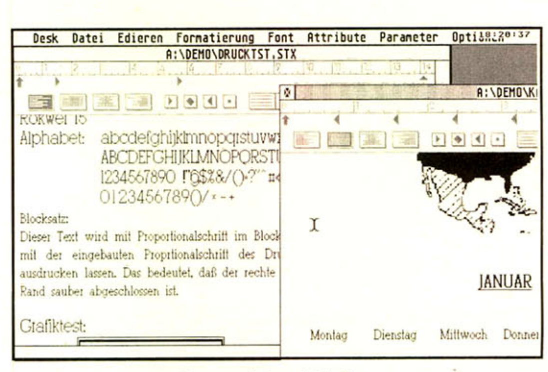
DMC zeigte die Version 1.64 von 'DynaCADD', das nun Schraffuren unterstützt. Ein besonderes Bonbon sind die Vektorzeichensätze, die mit diesem CAD-Programm bearbeitet werden können. Sie erweitern die Fähigkeiten von 'DynaCADD' in Richtung grafischer Gestaltung deutlich.

Noch eine Neuigkeit im CAD-Markt kommt von **Technobox** aus Bochum. 'Technobox CAD/2' lautet die korrekte Bezeichnung des weiterentwickelten 'Campus CAD' bzw. 'Techno-CAD 1.3.' (Welche Namensvielfalt!). Beeindruckend ist die Funktionsübersicht, anhand derer sich zeigt, daß die Version 2 doch weit mehr als nur ein Upgrade ist. Präsentiert wurde sie interessanterweise auf einem ATARI PC, denn die Versionen sind für ST und IBM-kompatiblen PC absolut identisch. Einzig die Farbfähigkeit fehlt der ST-Hardware. Technobox will mit dem neuen Produkt einmal mehr seine Bedeutung für den professionellen Markt unter Beweis stellen. Für den Hobby-Anwender hört beim Preis spätestens der Spaß auf: 'Technobox CAD/2' kostet 3498 DM.

Bei den Grafikprogrammen ging es nicht nur um Mal- und Zeichenprogramme oder DTP-Anwendungen, sondern auch (für Geschäftsleute ungemein wichtig) um Programme, die in der Lage sind, Präsentationsgrafiken zu erstellen. Ja, Sie wissen schon, diese Balken- und Tortendiagramme, die Ihnen signalisieren sollen, wie gut oder schlecht Ihre Geschäfte laufen. SCIGRAPH von der Hamburger Firma **Scilab** ist ein Vertreter solcher Programme. Das Programm erstellt vollautomatisch bis zu 24 verschiedene Basisgrafiken, entweder aus bereits vorhandenen Daten oder welchen, die mittels eines Editors eingegeben werden können. Da es sich um Vektorgrafiken handelt, kann man sie beliebig manipulieren. Jede Linie, jede Fläche sowie Texte können bewegt, kopiert oder verschoben werden. Intern werden die Grafiken in einem Raster von 32767x 32767 aufgebaut und bei der Ausgabe auf die verschiedenen Auflösungen angepaßt. Beim eingebauten Editor handelt es sich um einen Spalteneditor, der sehr einfach zu bedienen ist und viele Features besitzt, die die Eingabe von Daten sehr erleichtert. Er besitzt zwei eigene Fenster, so daß der Austausch von Daten sehr einfach zu bewerkstelligen ist. Bei der grafischen Ausgabe auf den Bildschirm stehen vier Fenster zur Verfügung, die man nach GEM-Manier vergrößern und verschieben kann. Bei der Ausgabe auf einen Drucker kann der Anwender zwischen 9-Nadel-, 24-Nadel- oder ATARI-Laserdrucker wählen. Bei einem so umfangreichen Produkt ist ein Preis von DM 599,- nicht zuviel.



SciGraph präsentierte Geschäftsgrafiken.



Script ähnelt WriteNow auf dem MAC.

Textverarbeitung...

... bleibt nach wie vor eines der heißesten Themen. So liefen auch über diese ATARI-Messe zahlreiche Besucher in der Hoffnung, diesmal DAS Programm zu finden, das alle ihre Wünsche abdeckt. Gezeigt wurde einiges, lieferbar ist davon wenig, schauen wir uns die neuen an:

Von **Applications Systems** aus Heidelberg kommt neben 'Signum!' nun ein weiteres Textprogramm. 'Script' findet sein Vorbild bei 'WriteNow', das auf dem Apple Macintosh läuft. Gleichwohl ist es vom 'Look and Feel' ein reinrassiges ATARI-Programm. Einige Features: Verwendung von Signum!-Fonts oder den im Druck schnelleren Drucker-Fonts, On-Line-Formatierung des Textes, Fußnotenverwaltung, Grafikeinbindung. Verschiedene Zeilenabstände und Fonts sind in einem Text nutzbar. 'Script' wird vorraussichtlich Mitte Oktober erscheinen und kostet 198 DM. Ferner war eine arabische Version von Signum! zu bewundern, die doch die Anpassungsfähigkeit der Heidelberger Software-Schmiede demonstriert.



Trubel auf dem ASH-Stand

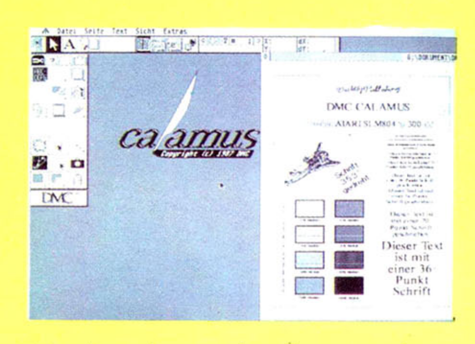
Nun ist es da: **'Word Perfect'** erscheint in einer Version für den ATARI ST. Die Version entspricht vom Funktionsumfang her der mit der Nummer 4.1 auf dem PC und heißt folgerichtig auch so. 'Word Perfect 4.1' ist zwar ein Produkt der Word Perfect Corporation, wird aber von ATARI selbst vertrieben werden. Eine englische Version steht kurz vor der Auslieferung; ab Ende Oktober soll es die deutsche zum Preis von 700 DM geben. Ein erster Blick auf dieses weithin bekannte Produkt bestätigt, daß es sich um keine bloße Anpas-

Grafik in neuer Dimension



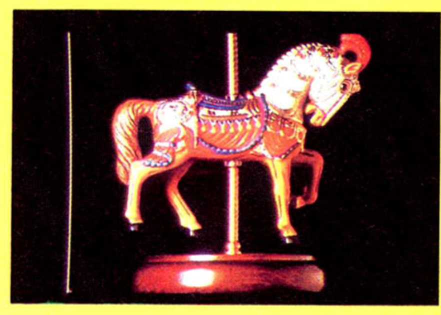
Die Tage, die für die Mega ST- Anwender nur schwarz und weiß sind, haben endgültig ein Ende gefunden. Eine neue Generation von hochauflösenden, farbenprächtigen Mega STs ist geboren. Die MAXON Graphic Expansion (MGE) verleiht dem ATARI Mega ST neues Leben. Viele und wichtige Anwendungen werden erst durch ein leistungsfähiges Grafiksystem sinnvoll. Das besaß bis heute kein Mega ST.

Die MGE bietet eine Lösung für Anwendungen, die nach höherer Auflösung und/oder vielen Farben verlangen. Mit seiner Vielfalt an verschiedenen Auflösungen wie zum Beispiel 256 gleichzeitig darstellbaren Farben bei einer Auflösung von 800 x 600 Bildpunkten oder 1280 x 1024 Bildpunkten bei maximal 16 Farben verwandelt sich Ihr Mega ST in eine richtig professionelle



Maschine, die man überall einsetzen kann, nicht nur bei CAD- und DTP-, sondern auch bei Video- und Grafikanwendungen.

Aber nicht nur bezüglich der Farbe hat die MGE immer eine Lösung parat; auch für den unverbesserlichen Schwarzweißanwender bietet die MAXON Graphic Expansion eine enorme Verbesserung. Jetzt können Monochromanwender bei einer Auflösung von 1280 x 960 Bildpunkten mit einem Großbildschirm arbeiten. Das ist aber immer noch nicht alles. Durch



einen der modernsten Grafikprozessoren und 1 MB Bildspeicher ist die Zeichengeschwindigkeit trotz höherer Auflösung und wesentlich mehr Farben um ein Vielfaches gestiegen.

Dem Programmierer liefern wir die notwendigen Werkzeuge, um mit der MGE selbst experimentieren zu können. Dem Anwender stellen wir unseren VDI-Treiber zur Verfügung, mit dem jedes sauber programmierte GEM-Programm sofort läuft. Dazu brauchen Sie nichts anderes als einen geeigneten Monitor (z.B. Multisync), und plötzlich ist alles viel größer und schneller. Die MGE wird einfach auf den internen Bus des Mega STs gesteckt. Fertig!

Auflösungen:

1. 13,75 MHz, 800 x 600 Bildpunkte, maximal 256 Farben interlace, 50 Hz Halbbildfrequenz (für Video- und Genlock-Anwendungen);
2. 27,5 MHz, 640 x 480 Bildpunkte, maximal 256 Farben aus 262144 non-interlace, 66 Hz Bildfrequenz;
3. 27,5 MHz, 800 x 600 Bildpunkte, maximal 256 Farben aus 262144 interlace, 80 Hz Halbbildfrequenz;
4. 55 MHz, 896 x 684 Bildpunkte, maximal 16 Farben aus 262144 non-interlace, 67 Hz Bildfrequenz;
5. 55 MHz, 1280 x 1024 Bildpunkte, maximal 16 Farben aus 262144 interlace, 65 Hz Halbbildfrequenz;
6. 110 MHz, 1280 x 960 Bildpunkte, monochrom non-interlace, 60 Hz Bildfrequenz.

Grafikprozessor Intel 82786

- Zeilenfrequenz, Bildfrequenz und Auflösung frei einstellbar;
- leistungsfähiger Display-Prozessor mit Hardware-Windowing-Fähigkeiten;
- leistungsfähiger Zeichenprozessor mit High-Level-Zeichenkommandos;
- Zeichengeschwindigkeit: 24 Millionen Bits/s bei Bit-Block-operationen;
- 25000 Zeichen/s;
- schnelle Kreisinterpolation;
- Unterstützung beim Erstellen von mit Mustern gefüllten Polygonen

Die MGE wird mit folgender Software geliefert:

- Hardware-Treiber
- Bindings für TURBO-C
- Bindings für Assembler
- GDOS-VDI-Treiber



Bestellcoupon MAXON Computer GmbH Industriestraße 26 6236 Eschborn Tel.: 06196/481811

Name: ______________________

Vorname: ______________________

Straße: ______________________

Ort: ______________________

Unterschrift: ______________________

Hiermit bestelle ich:

☐ Exemplare von "MAXON Graphic Expansion" für DM 1798,00

☐ Exemplare von "Entwicklerhandbuch MGE" (wird beim Kauf angerechnet) für DM 15,00

Vertrieb in der Schweiz: DTZ DataTrade AG Langstrasse 94 Postfach 413 CH-8021 Zürich Tel.: 01/242 80 88 Fax.: 01/291 05 07
Vertrieb in Österreich: Dipl.-Ing. Reinhart Temmel Ges.m.b.H. & Co.KG Markt 109 A-5440 Golling Tel.: 06244/7081-17 Fax.: 06244/7188-3
Vertrieb in Frankreich: AROBACE 2 Rue Piémontésis F-75018 Paris Tel: 1/42235044 Fax 1/42545631

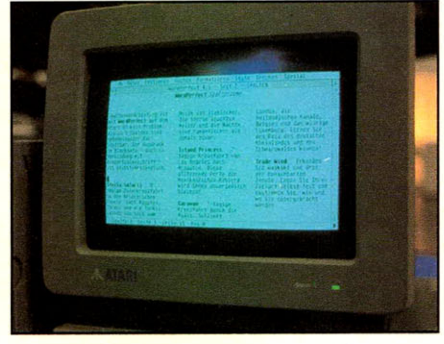
sung der PC-Version handelt: Das Textprogramm ist vollständig in GEM eingebunden und wird über Maus oder Tastatur bedient. Durch interessante Features wie Spaltensatz oder PostScriptausgabe wird es sicher Freunde auch auf dem ST finden.

Tempus Word fristet weiterhin sein Dasein als Ankündigung. **CCD** versprachen erste Exemplare - wie bereits auf der CeBIT - für die Jahreswende 89/90. Hoffen wir, daß es in den Händen der Entwickler aus Eltville bis dahin zur Serienreife gedeiht.

Lieferbar hingegen ist 'PKS-Write' von **DMC**. Die Leute aus Walluf bieten damit nach 'Calamus Online' das nächste Utility an, das Calamus' Ruf als komplettes DTP-System unterstreichen soll. Das besondere an diesem Texteditor ist, daß er wahlweise als Accessory oder Programm läuft. PKS-Write ist nicht als Textverarbeitung zu verstehen, es läßt auch keine Textformatierungen zu. Alle seine Funktionen sind vollständig darauf abgestimmt, den Text nach der Eingabe mit Calamus weiter zu verarbeiten. Das reicht soweit, daß sogar Format-, Stil- und Font-Anweisungen, also Satzinformationen für das DTP-Programm, mit dem Text eingegeben werden können. 'PKS-Write' kostet 198 DM.

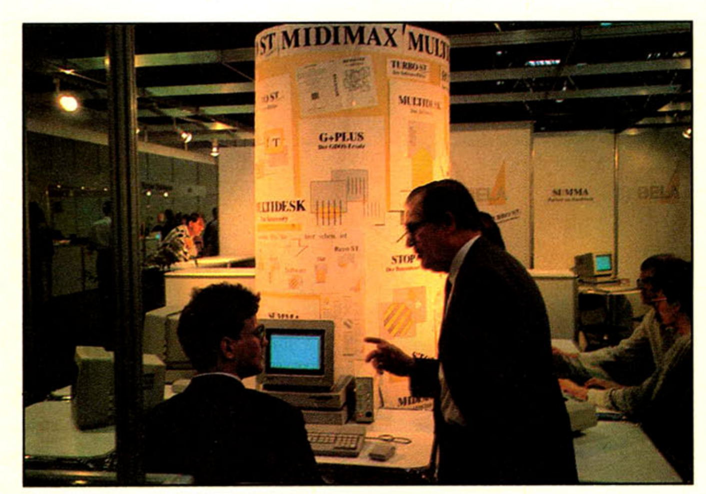
Stricken und Emulieren

Eine kleine Attraktion in der Halle 1, deren Schwergewicht bei Spielen und Musik lag, war eine Kombination aus ATARI ST und einer Strickmaschine von Brother. Die Brüder **Gunther und Dietmar Schröder** haben ein Programm entwickelt, mit dem mühelos Strickmuster in den ATARI eingegeben und zu einer Strickmaschine gesendet werden können. Dabei werden zwei Farben (Grundfarbe und Muster) verwendet. Zusätzlich ist das Einlesen von Mustern mittels Scanner möglich. So können binnen kürzester Zeit aus gedruckten Vorlagen fertige Pullover werden. Das Programm 'pro Fashion 2000' kostet ohne Hardware 849 DM.



Word Perfect wird von ATARI vertrieben.

Bei **ADI** wurde die neueste Version des Datenbanksystems 'Adimens ST' gezeigt, das jetzt die Versionsnummer 3.0 und den Zusatz 'plus' trägt. Wichtigste Neuerungen: Neben Dateien und Verbindungen unter ihnen gibt es den sog. Verbund. In einer Verbundmaske werden Einträge aus verschiedenen Dateien zusammengefaßt. Ein Verbund wird bearbeitet wie eine virtuelle Datei. Kommentare können nun in die Masken eingefügt werden, und endlich ist Mehrfachsortierung möglich. Ab dem 15. Oktober '89 ist Adimens 3.0 lieferbar. Das Update von 2.3 wird 150 DM kosten, wer sein Adimens vor dem 31.12. '88 gekauft hat, wird von Adi mit einem 'Treuebonus' belohnt und bezahlt nur 100 DM.



Ein Blick auf den Bela-Stand


Saubere Maschen mit dem ST

Bei **Bela** schreitet die Entwicklung langsam, aber stetig voran. Der Echtzeit-Codierer STOP läßt sich nun einfacher bedienen und unterstützt maximal 12 Paßwörter. TURBO ST, den Softwareblitter, fanden wir in der Version 1.6 vor - das Ding ist einfach atemberaubend schnell geworden. Während in früheren Versionen nur die GEM-Textausgabe beschleunigt wurde, waren nun einige weitere Teile der LINE-A-Routinen dran. Das Ergebnis: Fenster im Desktop öffnen und schließen sich blitzartig, und in Wordplus läßt sich dieser Text nun fast angenehm schnell schreiben und umherscrollen. Ein Hoch dem Programmierer!

SUMMA, das Handwerkerpaket, nähert sich nach beinahe zweieinhalb Jahren Entwicklungszeit der Perfektion. SUMMA besteht aus einem Texteditor, einer Kundendatenbank und den Artikeldaten; zusammen mit einer Lohnabrechnung und einer Buchhaltung ergibt das ein komplettes Paket für mittelständische Betriebe. Das Programm kommt weitestgehend ohne Kundennummern aus; es hilft bei Angebotserstellungen, Ausschreibungen, Auftragsbearbeitung, Rechnungen und Mahnungen sowie bei der Kalkulation. BELA plant das Paket allerdings erst dann in den Handel zu bringen, wenn die Unterstützung des Anwenders durch geschulte Händler gewährleistet ist.

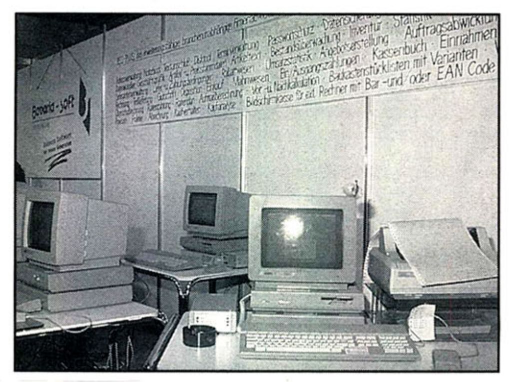
Weiterhin war 'Multidesk', ein Accessory, zu bewundern, das es erlaubt, bis zu 32 weitere Accessories nachzuladen und zu benutzen. 'G+Plus' heißt der Ersatz für GDOS, das ja dafür berüchtigt ist, die Bildschirmausgabe des ST zu verzögern. Das Utility verwaltet zusätzlich die Pfade zu den Gerätetreibern und Fonts der einzelnen Programme selbständig, so daß das Booten beim Wechsel zwischen Programmen, die GDOS benötigen, ebenfalls entfällt. 'Hotwire' ist eine Shell, die im Hintergrund des Desktops auf Tastendruck wartet, um automatisch Programme oder ganze automatisierte Abläufe zu starten. 'Midimax' schließlich ist eine Hilfe für Midi-Anwender. Wahlweise als Programm oder Accessory zu benutzen, erlaubt es, auf einfachste Weise Befehle an ein Midi-Gerät zu senden. Makros können definiert und auch durch Midi-Ereignisse getriggert werden.

Ein neuer Stern am Midi-Himmel heißt **Mopro** und kommt aus Holland. Diese Firma zeigte ein System zur Ein- und Ausgabe beliebiger Klanginformationen über den ST. Die Sample-Hardware 'Son' arbeitet mit 16 Bit und einer Frequenz von 44,1 kHz, also voller CD-Qualität. Das System ist in der Lage, die Daten in Echtzeit auf Harddisk(s), die an den ST angechlossen sind, zu schreiben. (Vier Minuten Musik belegen ca. 44 MB Speicherplatz). Die dazu gehörende Software 'Solfege' erlaubt das einfache und vielfältige Edieren der eingelesenen Stücke und natürlich die entsprechende Ausgabe. Mit ca. 4400 DM ist dieses System allerdings keine Low-Cost-Lösung mehr.

Die Heilbronner Firma **Lighthouse** ist bekannt dafür, dem ATARI stets ein neues Äußeres zu verschaffen. Sie zeigte auf der Messe ein Tower-Gehäuse, in dem ein Mega ST nebst Festplatten und Floppy-Laufwerken problemlos Platz findet. Nachdem also der Rechner unter dem Tisch verstaut ist, gilt es, den Arbeitsplatz weiter aufzuräumen. Hier hilft Lighthouse mit einem Moni-

tor-Schwenkarm, auf dem auch die Tastatur eine Ablage findet.

Ab sofort läuft die Firmenabwicklungs-Software 'BSS-Plus' auch im Netzwerk. **Bavaria-Soft** aus Ottobrunn zeigten auf der ATARI-Messe zum ersten Mal die Netzwerkversion ihres Software-Pakets. Bis zu 20 STs können über die Netzwerke von BioNet oder eLAN gemeinsam auf den Datenbestand eines Servers Zugriff nehmen. Dieser Server kann durchaus auch eine MS-DOS-Maschine sein. Da das gesamte 'BSS-Plus'-System modular gehalten ist, ist es schwer, einen Preis für eine eine solche Firmenlösung zu nennen. Trotzdem ein Beispiel: Für ein Netzwerk mit zwei Arbeitsplätzen und einem ST-Server nebst entsprechender Software und kompletter Hardware (Rechner und Netzwerk, aber ohne Drucker) sind ca. 27000 DM zu veranschlagen.


BSS-Plus wurde netzwerkfähig

Das waren drei Tage Messe, auf denen es viel zu sehen und zu hören gab. Wie Claus Brod in seinem Editorial schreibt, wäre es für die Menge an Besuchern, den Aufwand und die Vielzahl an neuen Produkten, die jedes Jahr vorgestellt werden, wünschenswert, wenn die Messe mindestens einen Tag länger dauern würde. Vielleicht sehen wir uns nächsten Sommer in Düsseldorf am Donnerstag um 9 Uhr. Bis bald.

Letzte Meldung

In letzter Minute haben wir erfahren, daß ATARI ab Anfang Oktober die lang erwartete **TOS-Version 1.4** ausliefern wird. Wer schon einmal das Vergnügen gehabt hat, diese Version auf einem Farbmonitor laufen zu sehen, der kann sicherlich verstehen, warum der offizielle Name von TOS 1.4 jetzt *Rainbow-TOS* lautet. Rainbow-TOS wird **nicht** serienmäßig mit den STs geliefert werden, sondern nur quasi als Ersatzteil beim ATARI-Händler für DM 198,- verfügbar sein. Da nachträglich noch zwei Fehler gefunden wurden, wird ein Patch-Programm für den Autoordner mitgeliefert werden.

CB/IB/MM/HE



Auf der ATARI-Messe '89 wurde der neue 1040 ST^{E} vorgestellt. Was kann er? Was kann er nicht? Ist er kompatibel zum "alten" ST? Wir berichten von unserem Testgerät!

Mit dem ST^{E} scheint ATARI dem AMIGA Konkurrenz machen zu wollen. Die durchaus überzeugenden technischen Daten dieses Geräts lassen es jedenfalls vermuten. Der ST^{E} soll Auge und Ohr verwöhnen. Dazu bietet er Stereo-Sound in CD-Qualität, erzeugt von einem 8-Bit-PCM-Generator ("Pulse Code Modulation), der endlich auch den letzten AMIGA-Freak vor Neid erblassen lassen müßte. Dieser Generator ist übrigens ebenfalls im TT eingebaut worden. Doch auch in Sachen Grafik kann sich "der Neue" sehen lassen: 16 von 4096 möglichen Farben lassen sich darstellen, und das bei einer Auflösung von 320 x 200 Pixeln im ATARI (teilweise wurde fälschlicherweise von 512 Farben aus 4096 berichtet). Das wird dadurch möglich, daß die Farbpalette um 3 Bit erweitert wurde, jeweils eines für die Farbinformationen Rot, Grün, Blau (RGB). Eine große Wirkung durch eine kleine Maßnahme! Natürlich bleibt der gewohnte 640 x 400 Pixel-Monochrom-Modus erhalten. Dadurch läßt sich auch der SM 124 weiterhin betreiben. Mit diesen Fähigkeiten scheint der ST^{E} wie geschaffen für Spiele und siehe da: Ihm wurden vier weitere Joystickports spendiert, also insgesamt sechs (!) Anschlußmöglichkeiten für Joysticks, Mäuse, Paddles und Lichtpistolen. Ob dies ohne Multitasking-Betriebssystem so sinnvoll ist, ist eine andere Frage. Für den Anschluß der vier weiteren Joysticks wird ein Adapter benötigt, da an einen Port jeweils 2 Joysticks angeschlossen werden. Damit der Spielspaß ungehindert seinen Lauf nehmen kann, ist es nun *endlich* möglich, ruckfrei horizontal und vertikal zu scrollen: hardwaremäßig! Dazu wurde eine neue Adresse im Videoshifter eingeführt, über die beispielsweise das horizontale Scrolling gesteuert werden kann. Eine weitere gute Neuerung beim Bildschirm: Er kann jetzt an eine beliebige Wortadresse gelegt werden. So kann man sich den Speicher so aufteilen, wie man es gerne hätte. Um das Abspielen der (Stereo-)Hintergrundmusik muß sich der Programmierer nicht mehr kümmern - es existiert eine Routine im Betriebssystem, die diese Aufgabe interruptgesteuert erledigt, nicht einmal der Prozessor wird dadurch belastet, da hier mit DMA ("direct memory access") gearbeitet wird.

Anwendungen

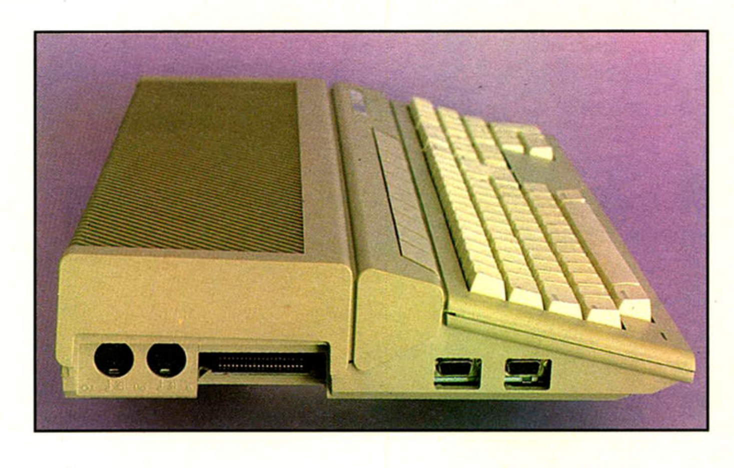
Fast die gesamte Palette der ST-Software kann auch auf dem ST^{E} benutzt werden. Nachdem wir unser Testgerät an eine Festplatte angeschlossen hatten, gaben sich alte Bekannte ein Stelldichein: 2 Bomben. Nachdem jedoch der Festplattentreiber (Eickmann) gegen den Allround-Treiber von Claus Brod ausgewechselt wurde, funktionierte alles einwandfrei. Natürlich funktioniert der Original-ATARI-Harddisk-Treiber ebenfalls ohne Probleme. Die eingebaute TOS-Version hat die Versionsnummer 1.6 mit einem Systemdatum vom 22.6.1989 und dürfte eine an den ST^{E} angepaßte Version des neuen TOS 1.4 ("Rainbow-TOS": Wenn man "Desktop Info" in Farbe anklickt, schimmert das

ATARI-Zeichen in vielen Farben) sein. Mit dem TOS werden die Zugriffe auf die Festplatte durch einen integrierten Cache-Speicher erheblich beschleunigt. Dadurch werden solch subtile Programme wie “Turbodos” überflüssig: Der ST^{E} arbeitet auch so schnell genug. Im Monochrommodus konnte kein ST-Anwenderprogramm gefunden werden, das nicht auch auf dem ST^{E} läuft. Getestet wurden u.a. Wordplus (3.x), GFA-BASIC, OMIKRON.BASIC, BECKERpage ST, Interlink, MagicBOX ST, Turbo C u.a. Anders verhält es sich aber zum Teil bei Spielen, da manche Spieleprogrammierer der Meinung sind, daß es vollkommen egal ist, ob man an eine Wortadresse ein Langwort schreibt. Bisher hatte dieses Überschreiben noch keine Folgen, doch neuerdings wird ein Byte, das überschrieben wurde, genutzt. Wir wollen dieses Verhalten allerdings nicht verallgemeinern, da z.B. Spiele wie Starglider II, Flight Simulator II, Tetris, Kick Off und andere einwandfrei funktionierten.

Zutaten

Auch die Innereien des ST^{E} können sich sehen lassen. Der MC68k-Prozessor ist (natürlich) noch vorhanden, allerdings einem gesockelten, quadratischen gewichen. Der Glue und die MMU wurden in einen (Fat-)Chip zusammengefaßt. Dieser ist leider in SMD-Technik festgelötet und dürfte, sofern er einmal defekt ist (was bei der MMU nicht selten ist), ein kleines Vermögen kosten. Ein näherer Blick auf den Soundchip enthüllt, daß noch immer der gute, alte Yamaha YM 2149 tätig ist. Dieser verrichtet seinen Dienst jedoch nur der Kompatibilität zum ST wegen: Die Platine wurde (wie bereits oben erwähnt) um den 8-Bit-PCM-Soundprozessor erweitert, welcher in seinen Funktionen tatsächlich (und endlich) mit dem des AMIGA zu vergleichen ist.

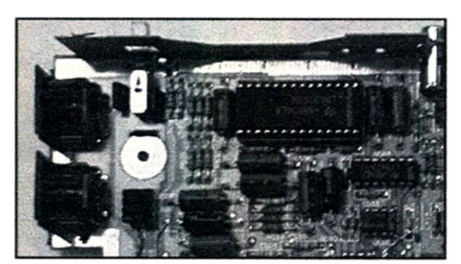
Nach dem Systembus sucht man vergeblich; er ist nicht vorhanden. Geschickt ist man bei eventuellen Speichererweiterungen vorgegangen: Der ST^{E} kann mit SIMM-Steckkarten, die auch schon beim Mac Verwendung finden, um jeweils 1 Megabyte erweitert werden. Sollte also ein Chip auf einer Steckkarte defekt sein, kann man getrost die gesamte Karte dem Mülleimer anvertrauen, da man beim Auslöten alle anderen Chips mit Sicherheit beschädigen würde. Als Laufwerk wurde ein Epson-Laufwerk eingebaut. Ohne mich beschweren zu wollen, meine ich, daß man die Wahl doch hätte geschickter wählen können. Erfreulich:



Die neue Lage der Midi-Buchsen und der zusätzliche Joystickport

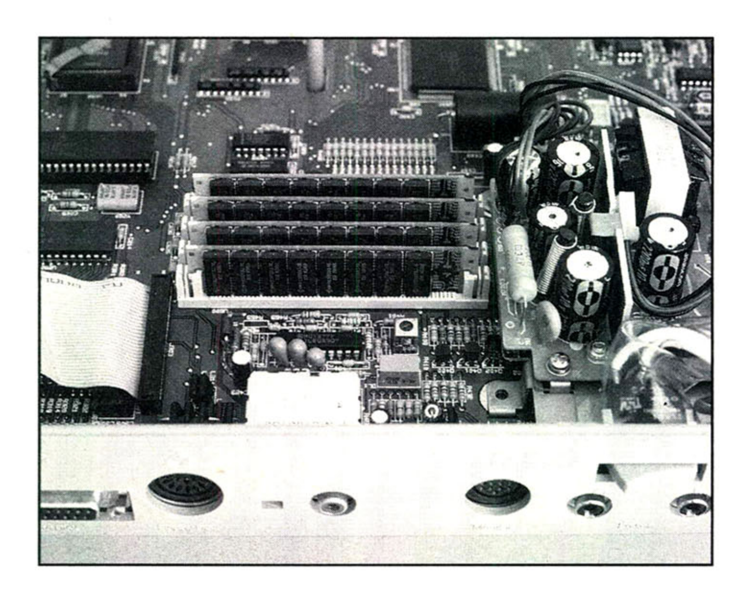
Stereo-Ausgänge
HF-Emulator
Midi-Buchsen
4 MB RAM auf SIMM-Bänken
TOS 1.6-Chips
Zusätzliche Joytick-Buchsen
FAT-Chip für MMU und GLUE
Blitter
CPU 68000

Dem Laufwerksschacht wurde endlich eine Staubschutzklappe spendiert, die ungebetenen Schmutz und Zigarettenqualm fernhält, sofern keine Diskette eingelegt ist. Hin und wieder scheint also auch ein Entwickler mitzudenken! Leider wurde das Gehäuse des 1040 ST übernommen, so daß auch die Nachteile dieses Rechners in Kauf genommen werden müssen: Um die Maus zu befestigen, muß man sich wieder im Fingerhakeln üben, da sie, 1040-üblich, unter dem Rechner in einen engen Schacht bugsiert werden muß. Hier werden viele Frauenfingernägel ihr Leben lassen müssen. Warum in aller Welt konnte man dieses Manko nicht ändern? Auch bei den neuen Joystick-Anschlüssen liegt einiges im Argen: Um vier Joysticks an zwei Ports anschließen



Der PCM-Generator sorgt für den richtigen Sound.

zu können, wird ein Adapter benötigt. Warum dieser Adapter nicht gleich eingebaut wurde, bleibt ungeklärt: So jedenfalls liegt ein Kabel mehr im Weg. Zur Abwechslung gibt es auch noch etwas erfreuliches zu vermelden: Der Videoshifter ist nun endlich über die Monitorbuchse synchronisierbar. Dadurch werden Genlock-Anwendungen überhaupt erst sinnvoll möglich, da nun endlich das Computerbild mit dem Videobild extern synchronisiert werden kann. Im Bereich Grafik hat man den Käufern noch etwas gutes beschert, was bislang nur den Mega ST-Benutzern vorbehalten war: Der gute, alte Blitter verrichtet hier (langsam wie eh und je) seinen Dienst. So wird dem von Natur aus etwas langsamen Desktop doch noch etwas auf die Sprünge geholfen (und die Kaffeemaschine kann ruhen). Die Midi-Buchsen liegen jetzt übrigens ebenso wie die zusätzlichen Joystick-Buchsen auf der linken Seite neben dem ROM-Port.



Die SIMM-RAM-Bänke im 1040 STᴱ

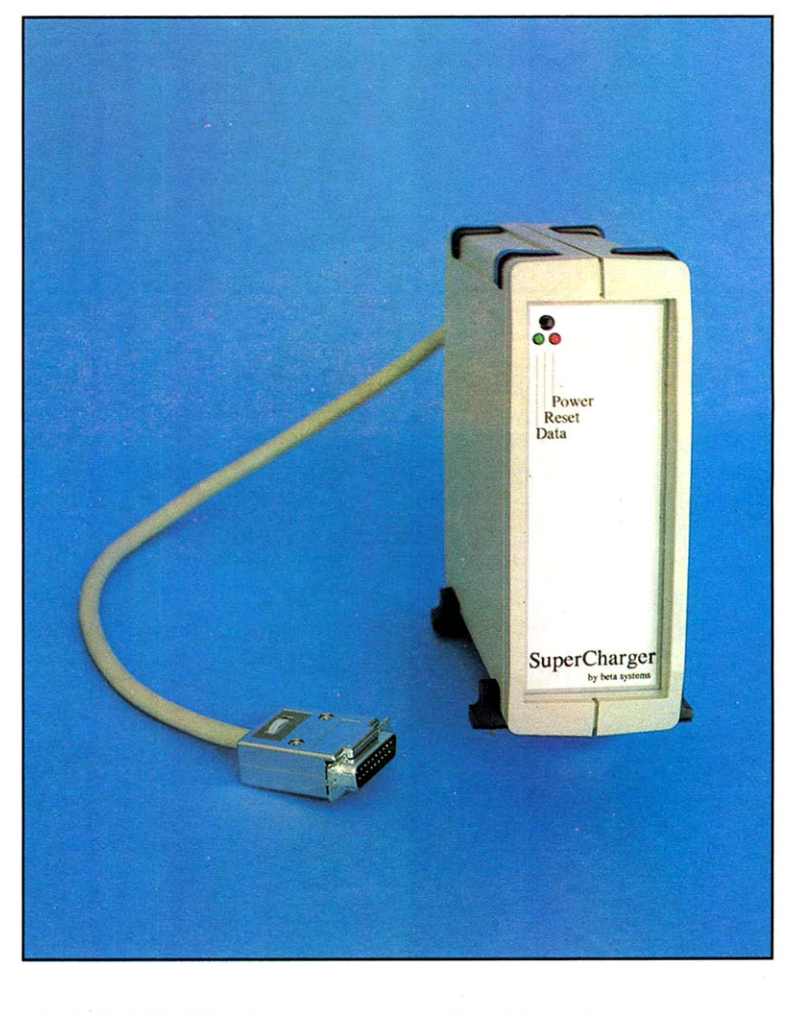
Also?

Wer sich heute einen ST kaufen will, sollte lieber warten bis der STᴱ lieferbar ist. Warum sollte man schließlich auf die Vorteile verzichten, die er bietet? Man kann getrost sagen, daß der STᴱ zu 99% kompatibel zum ST ist. Außerdem ist tatsächlich auch im Spielebereich eine gute Alternative zum Konkurrenten AMIGA geschaffen worden. Wer "siegen" wird, wird die Zukunft zeigen. Der offizielle Verkaufspreis für den 1040 STE mit 1MB und Monitor SM 124 liegt bei 1598,- DM. Die Auslieferung soll etwa sechs Wochen nach der ATARI-Messe stattfinden, also etwa Anfang Oktober. Laut ATARI sind die Geräte auch bereits lieferbar. Ein Besuch beim Händler lohnt sich also.

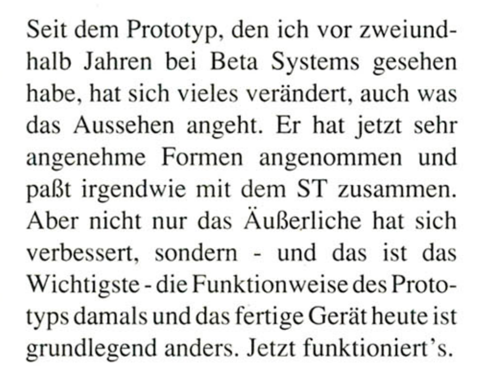
MP

SuperCharger

Das Ende einer Odyssee



Er wurde vor ungefähr zweiundhalb Jahren angekündigt, als MS-DOS-Emulatoren für den ST sehr gefragt waren, und es aber fast überhaupt noch nichts in dieser Richtung gab. Damals galt der SuperCharger als eine Sensation, nicht nur weil er eine einfache Brücke zwischen der MS-DOS- und der ATARI ST-Welt bildet, sondern auch wegen seinem versprochenen Leistungsumfang. Seit damals wurde viel darüber gesprochen und geschrieben. Er wurde sogar verkauft, obwohl er noch nicht fertiggestellt war. Die Geschichte des SuperChargers ist ein von vielen Kontroversen belasteter Vorfall. Jetzt ist er da! Nicht mehr als Phantom, das man nicht anfassen kann, sondern als serienfertiges Produkt.



Seit dem Prototyp, den ich vor zweiundhalb Jahren bei Beta Systems gesehen habe, hat sich vieles verändert, auch was das Aussehen angeht. Er hat jetzt sehr angenehme Formen angenommen und paßt irgendwie mit dem ST zusammen. Aber nicht nur das Äußerliche hat sich verbessert, sondern - und das ist das Wichtigste - die Funktionweise des Prototyps damals und das fertige Gerät heute ist grundlegend anders. Jetzt funktioniert's.

Das Bauen eines Emulators, der einen total anderen Rechner mit einem völlig verschiedenen Prozessor nachahmen soll, hat sich als sehr schwierig erwiesen. Mit diesem Problem müssen sowohl die Hersteller reiner PC-Clones als auch diejenigen leben, die einen funktionstüchtigen PC-Emulator für den ATARI-ST konstruieren. Bei der Entwicklung des SuperChargers war es mit Sicherheit nicht anders.

Der Mini-PC

Beim SuperCharger kann man mit Sicherheit sagen, daß es sich um einen Mini-PC handelt, der durch geschickte Konstruktion reibungslos mit dem ST läuft. Auf der Platine befindet sich eine NEC V30-CPU mit 8 MHz Taktfrequenz, die eine höhere Rechenleistung als ein INTEL 8086-Prozessor hat, aber vollständig kompatibel zu diesem bleibt. Ein von Beta Systems entwickeltes Gate-Array sorgt für die Kommunikation zwischen den Beiden. Ebenso sorgt dieser Baustein dafür, daß nach einem Reset ein kleiner Bootlader auf der ATARI-Seite den Emulator zum Leben erweckt.

In der Standardausführung verfügt der SuperCharger über ein halbes Megabyte Speicher sowie ab Mitte September (laut Vertriebsfirma SEH) über ein Microsoft-DOS in der Version 4.0. Auf Anschlußmöglichkeiten externer Peripheriegeräte wurde mit Absicht verzichtet, da der ATARI ST über diese schon reichlich verfügt. Auf der Platine befinden sich Steckplätze für einen arithmetischen Coprozessor 8087 (bei unserem Testgerät schon eingebaut) sowie für ein zusätzliches halbes Megabyte Speicher. Bei 512 kByte SuperCharger-Speicher stehen einem noch 384 kByte frei zur Verfügung (bei 1 MByte sind es 704 kByte). Anderes als vor zweiundhalb Jahren ist in der fertigen Version kein Platz für andere Erweiterungen vorgesehen. Wie gesagt, man bekommt mit dem SuperCharger einen kleinen PC, der problemlos an den ATARI ST anzuschließen ist. Problemlos deswegen, weil es sich bei ihm um eine steckbare Lösung handelt und keinerlei Eingriffe im Computer gemacht werden müssen. Man steckt ihn einfach in den DMA-Port, startet die Software und prompt meldet sich der SuperCharger auf dem Bildschirm. Für ATARI ST-Anwender, die eine Festplatte benutzen möchten, sei gleich erwähnt, daß der DMA-Port durchgeschliffen wurde. Die Harddisk läßt sich also ohne weiteres benutzen.

Geschwindigkeit ist angesagt

Eines der lästigen Probleme bei PC ditto, der reinen Software-MS-DOS-Emulation auf dem ST, ist die langsame Ausgabe auf dem Bildschirm. Für eine Software-Emulation war das verständlich. Der ST muß zuerst alles interpretieren, dann dem TOS schmackhaft machen (auf das passende Format bringen) und am Ende alles auf den Bildschirm bringen. Für die intensive Arbeit mit einer Textverarbeitung oder einem Editor kann der PC ditto dann auf Dauer auch leicht zu einer Tortur werden.

Beim PC-Speed, einem anderen Hardware-MS-DOS-Emulator, läuft die Sache absolut anders und deswegen sehr

schnell. Der Emulator und der ST greifen auf den gleichen Speicher zu, nämlich auf den internen Speicher des ST. Außerdem braucht PC-Speed nichts zu interpretieren, das erledigt die eingebaute V30-CPU. Eine sehr durchdachte Software bringt es sogar fertig, daß der Prozessor des PC-Speeds direkt in den Videospeicher des ST schreibt. Dadurch wird die Bildschirmausgabe unheimlich schnell. Ein Vergleich mit dem oben genannten PC ditto ist in diesem Bereich unmöglich.

Die Frage ist aber, wie schnell ist der SuperCharger? Auch hier wurde eine andere Lösung angewandt. Bei diesem Emulator wird der gesamte Bildschirm erst im eigenen RAM gespeichert (er besitzt ja mindestens 512 kByte), diese Bildinformation wird dann via DMA zum ATARI geschickt, der dann noch einiges zu kämpfen hat, bis die Zeichen auf dem Bildschirm landen. Das benötigt alles seine Zeit, ist aber trotz dieses Aufwands sehr schnell, so daß auch hier kein Vergleich zu PC ditto möglich ist. Um es Ihnen etwas deutlicher zu gestalten, kann man ruhig sagen, daß keinerlei Abstriche gegenüber einem mit 4,77 MHz getaktetem Original-PC-XT zu erwarten sind. Ebenso ist das Zeilen-Scrolling, das übrigens ständig verbessert wird, sehr schnell geworden, so daß sich z.B. mit einer Textverarbeitung gut arbeiten läßt.

SuperCharger verfügt nur über eine MGA- und eine CGA-Emulation. Die stark verbreitete hochauflösendere Hercules-Grafik wird nicht unterstützt, da sie schwierig nachzuahmen ist, weil die Grafikauflösung des ST dafür nicht ausreicht. PC Speed stellte bisher als Alternative Teilbereiche dar, die gescrollt werden konnten (Neuerdings ist mit einem Trick der ganze Bereich darstellbar.). Die Textausgabe ist beim SuperCharger durch selbstdefinierte Zeichen um einiges angenehmer als bei der Original-PC-Textausgabe. Auch bei der Farbdarstellung stößt man schnell an die Grenzen des ATARI ST. Bei der niedrigste Auflösung des STs stehen statt der 16 Farben im Vordergrund und den acht Farben im Hintergund, die ein Original-PC mit CGA-Karte anbietet, nur vier Farben zur Verfügung. Damit läßt sich aber leben. Auch auf dem Monochrommonitor läuft der Farbmodus. Hier werden die Farben als verschiedene Rastermuster dargestellt. Die Grafik ist gegenüber einem Original-Rechner ausgesprochen gut, da sie durch eine Verdopplung der Zeilenzahl erheblich verbessert wird.

Geschwindigkeitstest

Supercharger wurde von uns zwei Geschwindigkeitstests unterworfen: der Norton-Utility und dem Performance-Analyzer-Test. Hier sind die Ergebnisse:

```
µDESIGNS    IBM COMPATIBLE PERFORMANCE ANALYZER    (C) 1986
              Copyright (C) 1986 Richard B. Johnson

   Checking memory block write   ticks: 122 compared to IBM/PC 413%
   Checking register to memory   ticks: 212 compared to IBM/PC 231%
   Checking memory to register   ticks: 242 compared to IBM/PC 202%
   Checking register to register ticks: 222 compared to IBM/PC 248%
   Checking divide by register   ticks: 84 compared to IBM/PC 609%
   Checking divide by memory     ticks: 135 compared to IBM/PC 447%
   Checking multiply by register ticks: 104 compared to IBM/PC 462%
   Checking multiply by memory   ticks: 127 compared to IBM/PC 438%
   Checking stack operations     ticks: 224 compared to IBM/PC 198%
   Checking far jumps, far calls ticks: 259 compared to IBM/PC 200%

 Total time is: 1731 clock ticks, (96 seconds) compared to IBM/PC 297%

A>
```

Zum Vergleich haben wir drei andere Rechner und PC-Speed herangezogen, so daß sich folgende Tabelle ergibt:

	SuperCharger V30/8MHz	PC-Speed V30/8MHz	PC-XT 8088/4,77MHZ	PC-XT V20/4,77MHz	PC-AT 80286/8Mhz/32 Bit-CPU
Norton-Faktor	4,0	4,0	1	1,8	9,0
Performance-Test	297%	280%	100%	150%	497%

Eine graue Maus & ein stummer PC

Bei der Lösung der Mausansteuerung ist man bei Beta Systems einen Weg gegangen, über den man sich streiten kann. Damit man keine andere Maus zu kaufen braucht, hat man der ATARI-Maus den seriellen Port *COM1* zugewiesen, so daß dieser leider nicht mehr für DFÜ-Kommunikation oder serielle Drucker zur Verfügung steht. Die Maus funktioniert aber einwandfrei.

Probleme tauchen eher dann auf, wenn man versucht, dem SuperCharger Töne zu entlocken. Das kann er nicht. Es handelt sich hier um einen absolut stummen PC.

Unter dem SuperCharger läßt sich über die parallele Schnittstelle des STs auch ein Drucker ansteuern. Dieser wird MS-DOS-like unter dem Gerätenamen *LPT1* angesprochen. In Zukunft soll (laut Hersteller) ebenso der ATARI-Laserdrucker unterstützt werden.

Prüfstand

Wie Sie sicherlich bemerkt haben, ist bis jetzt noch kein größeres Problem aufgetaucht. Wir haben aber auch noch nicht über die wichtigsten Punkte gesprochen. Nachdem man den SuperCharger an den DMA-Port angeschlossen hat, fängt die Installation erst an. Das bedeutet, man muß dem System zuerst mitteilen, über welche Gerätekonfiguration man verfügt, also z.B. eine externe Floppy, ein Monochrommonitor oder eine Harddisk etc. Das alles wird durch ein Installationsprogramm, das zum Lieferumfang gehört, konfiguriert. Hat man dies vollbracht und verfügt in seinem SuperCharger über keinen arithmetischen Coprozessor, kann man das Programm starten, das die Bindung zwischen SuperCharger und ST schafft. Unter dem Namen *ABIO.TOS* versteckt sich das Elixier, das den Emulator zum Leben erweckt.

Ist alles ordnungsgemäß verlaufen, wird man aufgefordert, eine Diskette mit dem DOS-Betriebssystem ins Laufwerk zu stecken. Damit ist man fertig! Man könnte aber auch sagen: So fängt alles an!

Das MS-DOS-Betriebssystem läßt sich ebenfalls von einer Harddisk starten, womit der ganze Bootvorgang automatisiert wird. Die Installation ist recht einfach und sollte auch für einen nicht hundertprozentigen PC-Kenner kein Problem sein. Zwei Punkte muß man dabei beachten: Erstens muß der SuperCharger immer angeschaltet sein, da das Durchschleifen des DMA-Ports sonst nicht funktioniert und die Festplatte, auch wenn sie nur vom ATARI benutzt wird,

nicht mehr angesprochen werden kann. Der zweite Punkt ist, daß die Boot-Partition, die für DOS reserviert wurde, nicht unter TOS benutzt werden darf, da sonst mit Datenverlust zu rechnen ist. Es sind aber alle TOS-Partitionen unter DOS zu benutzen, womit ein Datenaustausch möglich ist.

Der SuperCharger ist installiert, MS-DOS ist auch geladen, jetzt ist die Zeit gekommen, festzustellen, welche Software mit dem Emulator läuft. Wir hatten - wie bereits erwähnt - diesen Emulator vor über zwei Jahren in seiner Entwicklungsphase gesehen. Damals wurde uns gezeigt, wie gut Turbo Pascal von Borland funktioniert. Das ist allerdings auch ein Programm, das normalerweise keine Schwierigkeiten bereitet. Es läuft bis jetzt mit allen MS-DOS-Emulatoren, die es mittlerweile für den ST gibt. Für einen 'harten' Test ist es fast ungeeignet. Wir haben es trotzdem gemacht, da die Borland-Produkte sehr beliebt und verbreitet sind. Egal wie Turbo Pascal funktioniert, bei SuperCharger war es tadellos. Die Geschwindigkeit ist ausgesprochen gut, und man vergißt sehr schnell, daß man vor einer Emulation sitzt. Den gleichen Effekt hat man bei Turbo BASIC und Turbo C, die problemlos mit dem SuperCharger funktionierten. Mit diesen Produkten testeten wir auch den 8087-Coprozessor und stellten keinerlei Probleme fest.

Es ist sehr schwierig, ein Geschwindigkeitsgefühl weiterzuvermitteln. Man kann z.B. sagen, Signum! ist in bezug auf Wordplus eine relativ schnelle Textverarbeitung, aber doch deutlich langsamer als Tempus. Die Frage ist also, ob man mit Signum! oder Wordplus arbeitet, also Briefe schreibt, Manuskripte anfertigt, Bücher setzt usw., ohne daß das Tippen zu einer Quälerei wird, weil der Rechner bzw. die Software nicht mithalten kann. Das ist z.B. teilweise ein Problem PC dittos, der einfach zu langsam ist, um mit ihm langfristig zu arbeiten. Bei SuperCharger ist das nicht der Fall. Man kann WordPerfect (MS-DOS) nehmen und damit um einiges schneller arbeiten, als mit Wordplus auf der ATARI-Seite. Das Zeilen-Scrolling ist schnell genug, und Funktionen wie Suchen oder Ersetzen sind weder langsamer noch schneller als auf jedem normalen PC-XT.

Man hat sich ebenso in Sachen Geschwindigkeit und Kompatibilität immer wieder den Flugsimulator von Sublogic vorgenommen. Er ist von seiner Grafik her sehr aufwendig und dank seiner Programmierung ebenso sehr kritisch auf jedem PC zu betreiben. Diesen Test hat der SuperCharger ebensogut bestanden. Nur der Ton kam nicht natürlich nicht zustande. (Wie oben erwähnt, handelt es sich ja um einen stummen PC.)

Ebenso arbeiteten Problemkinder wie die Norton Utilities oder Side Kick mit dem SuperCharger. AutoCAD, eines der beliebtesten CAD-Programme im PC-Bereich läuft ebenso unbeschwert. Sogar die Version 10.1, die einen arithmetischen Coprozessor verlangt, arbeitet mit dem SuperCharger, als wäre er ein echter PC. Auch hier ist die Geschwindigkeit bei aufwendigen Funktionen wie Zoomen oder Verschieben erstaunlicherweise sehr hoch. Die reinen Geschwindigkeitstests, die am PC unheimlich verbreitet sind, ergeben am SuperCharger ebenso gute Ergebnisse (siehe Tabelle).

Besonderheiten

Und als wäre dies nicht genug, stellt der SuperCharger dem Anwender noch ein paar Besonderheiten zur Verfügung. Einige sind schon vorhanden, andere sollen in Zukunft realisiert werden. Das wäre z.B. die Möglichkeit, den vorhandenen Speicherplatz, der auf der Platine des SuperChargers untergebracht wurde, als gepufferte RAM-Disk für den ST zu benutzen. Daran wird in Moment noch gearbeitet, und es sind auch schon Versuche damit erfolgreich gelaufen, es arbeitet aber noch nicht ganz fehlerfrei. Eine andere Möglichkeit, die auch in der Entwicklungsphase läuft, ist die Nutzung der V30-CPU als Coprozessor für den ATARI ST. Auch hier laufen schon bestimmte Programme. In die jetzige Version ist schon eine Art Freezer integriert, der das Umschalten zwischen dem ATARI- und dem SuperCharger-Modus ermöglicht. Nehmem wir mal an: Sie schreiben gerade eben in WordPerfect unter MS-DOS einen Brief, Sie haben aber diesen Brief schon vor langer Zeit unter TOS in Signum! angefangen, Sie möchten ihn jetzt aber gerne unter WordPerfect fertigstellen, doch der alte Text liegt nur im Signum!-Format vor, und damit kann WordPerfect nicht viel anfangen. Kein Problem! Sie betätigen ALT-CTRL-BACKSPACE zusammen und der ST wird neu gestartet. Dann rufen Sie Signum! auf, speichern Ihren alten Text dort im ASCII-Modus, starten dann wieder den SuperCharger und befinden sich wieder automatisch in Word Perfect, und zwar an der Stelle, wo Sie aufgehört haben. Von dort können Sie dann den ASCII-Text laden und weiterschreiben. Ist das nichts?

Fazit

Wie bereits anfangs gesagt wurde, ist SuperCharger ein Gerät das von seiner Vorgeschichte geprägt ist. Es wurde einfach zuviel darüber gesprochen, gewartet und erwartet. Heute, nach Betrachtung des fertigen Produktes, finde ich es ungerecht, wenn nur in Anbetracht dieser Vorgeschichte ein Urteil gefällt wird. Der Anwender bekommt heute für DM 798,- ein Gerät, das einfach anzuschließen ist und das tadellos funktioniert. Im Lieferumfang befindet sich auch ein MS-DOS 4.0, was ja aus Preisgründen auch nicht gerade üblich ist. Jetzt ist nur noch darauf zu warten, daß die Vertriebsgesellschaft mit der Auslieferung zurechtkommt und die Anwender nicht noch einmal enttäuscht werden.

Robert Wurm

Bezugsadresse:

SEH GmbH
Beethovenstr. 26
6455 Erlensee
Tel.: 06183/830

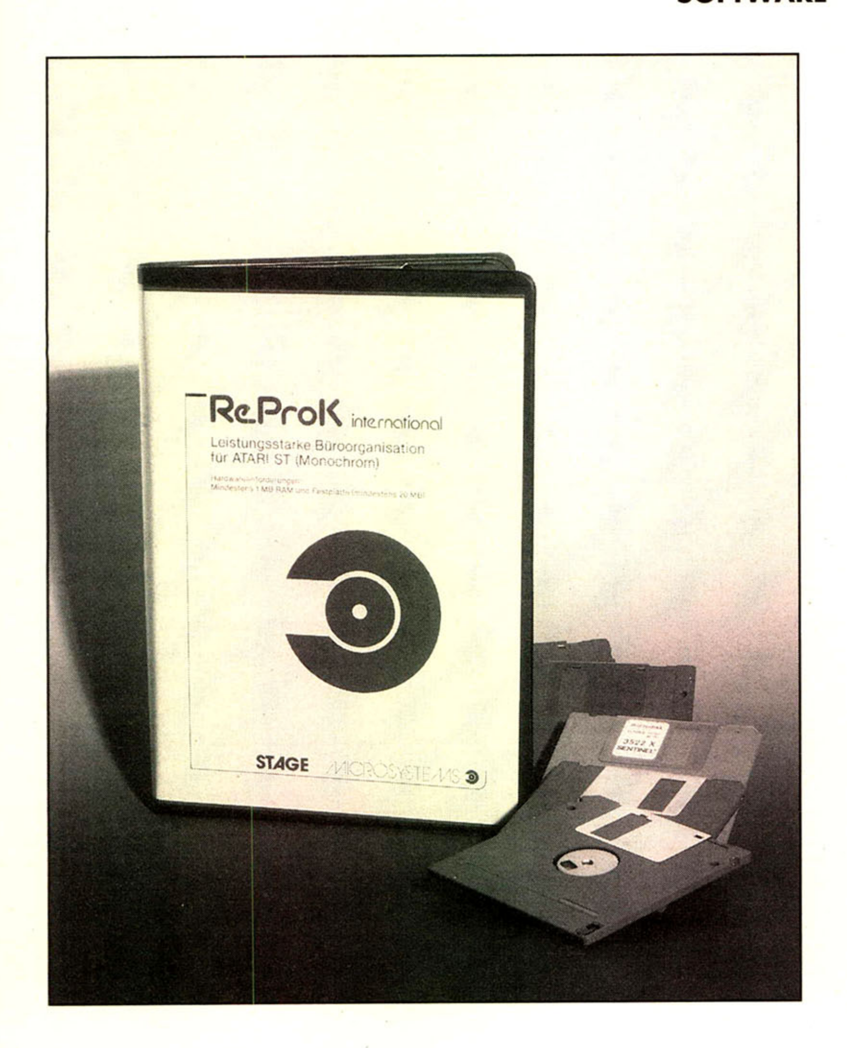
Bisher getestete Hard- und Software

Hardware

Alle ATARI ST-Rechner inkl. Mega ST
alle ATARI-Festplatten
alle Vortex-Festplatten
alle Eickmann-Festplatten

Software

MS-DOS-Versionen 3.2, 3.3, 4.01
MS Windows 1.02, 2.01, Windows 286
MS Word 3.0, 4.0
MS Excel
Arts & Letters
Aldus Pagemaker 2.0, 3.0
Symatec F+A 2.0, 3.0
DR GEM 3.0, 3.1
Ventura Publisher 1.00, 1.01
GEM Artline
PC Tools 4.1, 4.3, 5.1
(PC Secure, Compress usw.)
Norton Utility
Norton Commander
Freedom of Press
(Software-PostScript-Emulation)
Turbo-Pascal
GW-BASIC
Deluxe Paint II
Sublogic Flight Simulator 3.0
Sublogic JET



Büro organisieren mit ReProK

In meinem Bücherschrank schlummert ein etwas eingestaubtes Werk mit dem Titel "Hoch lebe die Organisation" (Parkinson). Dort werden vornehmlich Thesen für den effektiveren Ablauf des Arbeitsalltags aufgestellt. Ja, etwas veraltet ist das gute Büchlein schon, denn als der Autor sich 1968 daranmachte Formeln und Tips auch zum Wohle des Büromenschen aufzuschreiben, wurde an einen Siegeszug der elektronischen Datenverarbeitung noch kaum ein Gedanke verschwendet.

Heute sieht das schon etwas anders aus. Hard- und Software werden immer besser, schneller und (trotz alledem) auch immer preisgünstiger, so daß beispielsweise Inhaber von klein- und mittelständischen Betrieben auch schon gerne mit der Idee spielen, sich eine EDV-Anlage anzuschaffen. Besonders wenn die Ehefrau des Chefs abends den ganzen "Schreibkram" erledigen muß, steht der Computer ganz oben in der Bedarfsplanung.

Typisch Büro

Typische Büroarbeiten, wie Textverarbeitung, Adressverwaltung und Buchhaltung wurden aber anfangs fast nur Geräten der MS-DOS-Familie zugemutet, ganz einfach, weil dort die Softwareauswahl am größten war. Aber die Zeiten ändern sich! So ist der ATARI ST längst nicht mehr mit dem (völlig unberechtigten) Makel des Spielcomputers behaftet und erledigt vielerorts die Büroarbeit überaus zufriedenstellend. Allen Unkenrufen zum Trotz hat sich beispielsweise die GEM-Benutzeroberfläche als höchst effektiv, weil einfach zu erlernen und leicht zu bedienen herausgestellt.

Logischerweise muß es unter diesen Voraussetzungen doch auch Programme für jene typischen Büroarbeiten geben. Zugegeben, die Softwareanbieter zum ATARI ST haben sich bislang nicht gerade um "Büroprogramme" gerissen und dem Markt nicht das lukrative Potential zugetraut. Aber so allmählich kommen sie!

"Büroorganisation" ist der magische Begriff, der alles umschreibt, was bei vielen Kleinbetrieben noch mit "Zettelwirtschaft" gleichgesetzt wird. So fallen bei der täglichen Verwaltungsarbeit eine stattliche Menge an Daten an, die erfaßt, geordnet, kanalisiert, ausgewertet und abgelegt werden sollen. Wenn sich dann der Firmenchef dazu durchringt einen Computer anzuschaffen, dann soll er bitteschön einfach zu bedienen sein, und das Programm soll keine Rätsel aufgeben.

Auf der Düsseldorfer ATARI-Messe '89 wurde *ReProK* mit seinem Modul *Büroorganisation* einem breiten Publikum vorgestellt. ReProK hat es sich zur Aufgabe gemacht, (Originalton Werbeprospekt:) "... außerordentlichen Bedienungsluxus, flexible Einsatzmöglichkeiten und Preisattraktivität in einem Programm zu vereinen."

Nach dem Start

ReProK wird auf einer Diskette geliefert, ein Ringbuch mit 188 Seiten Beschreibung liegt bei. Zuerst muß eine Installationsprozedur durchlaufen werden, denn alle Programmteile, Masken und Daten liegen gepackt auf Diskette. Wenn nach dem Copyrightfenster die Desktopleiste erscheint, kann man vier Hauptarbeitsbereiche erkennen:

1. Der Verwaltungsblock

Dieser Programmteil wird oft auch Stammdatenverwaltung genannt. Dort sind Adressen von Kunden und Lieferanten, alle Produkte innerhalb von Warengruppen und einige "Feinheiten", wie länderspezifische Voreinstellungen, feste Textbausteine und Floskeln festgehalten. Der Vorteil einer solchen Stammdatenerfassung ist vielen EDV-Anwendern hinlänglich bekannt: Beispielsweise die Anschrift eines Kunden muß nur einmal

eingegeben werden und steht fortan für alle Vorgänge blitzschnell zur Verfügung.

2. Die Vorgangsverwaltung

Hier werden die kontinuierlichen Geschäftsvorfälle bearbeitet. Dabei wäre grob zwischen ausgehenden und eingehenden Vorfällen zu unterscheiden. Ausgänge sind alle Arbeiten (den Kunden betreffend) angefangen vom Angebot, über Auftragsbestätigung, Lieferschein und Rechnung, bis hin zur Kontrolle von Zahlungseingang, Gutschrift und Soll-Buchung. Eingänge sind im Gegenzug Lieferschein und Rechnung des Lieferanten, Zahlungsausgang und entsprechende Ist-Verbuchung.

3. Auswertungen

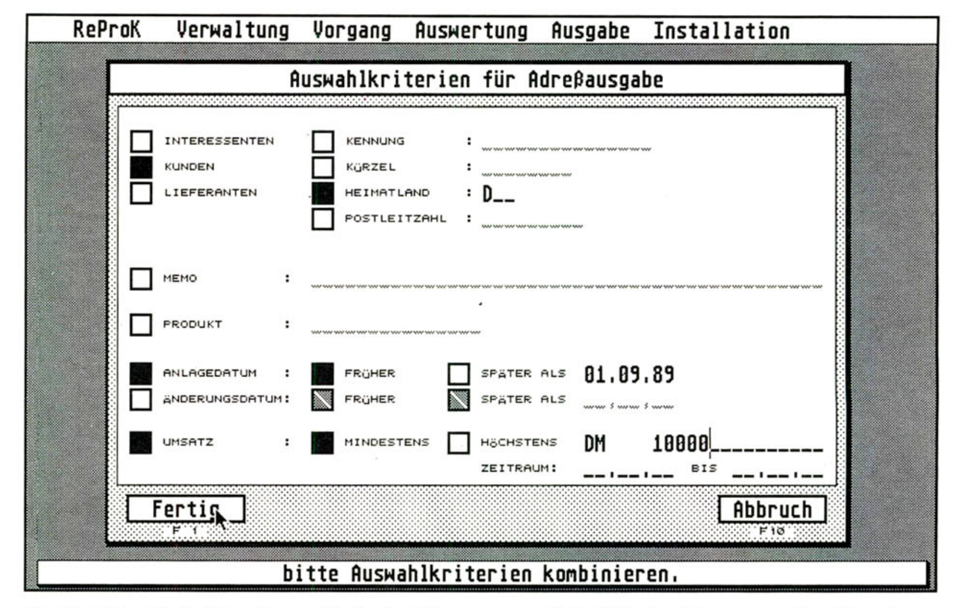
Es kommt sehr oft vor, daß Adressen nach Merkmalen sortiert werden müssen, beispielsweise um Kunden unter bestimmten Voraussetzungen anschreiben zu können. Gleiches kann mit Produkten passieren, um etwa die kalkulatorischen Verkaufspreise anzuheben. Auch sind hier die offenen Rechnungen (für Aus- und Eingänge) festgehalten.

4. Installation

Dieser Menüpunkt birgt hauptsächlich Anpassungen der verschiedenen Formulare an verschiedene Drucker. Es ist sogar denkbar, mehrere Drucker anzuschließen (sofern die Schnittstellen dazu vorhanden sind), jedem Drucker bestimmte Formulare zuzuweisen (also einer für Listen, ein anderer für Adreßaufkleber und ein weiterer für Briefe) und jeden Formularinhalt eigenständig zu konstruieren.

Sehen wir uns die Arbeitsweise doch etwas genauer an:

ReProK hat, dort wo es sinnvoll erschien, voll auf GEM-typisches Arbeiten mit der Maus gesetzt. Die Masken besitzen oft eine Vielzahl an Buttons zur Aus- und Anwahl verschiedener Funktionen. Wenn die Stamm- und Vorgangsdaten einmal alle eingetragen sind, kann für den regulären Arbeitsablauf fast völlig auf die Tastatur verzichtet werden (natürlich müssen Schlüsselzahlen oder -buchstaben weiterhin eingetippt werden). Wer es denn doch nicht lassen kann: Cursor- und Funktionstasten stehen nach wie vor (parallel zur Maus) zu Diensten, sogar mit einigen Ctrl- und Alt-Sequenzen können Menüs aktiviert werden. Nach einer kleinen Einarbeitungsphase läuft es mit der Maus aber recht flott.



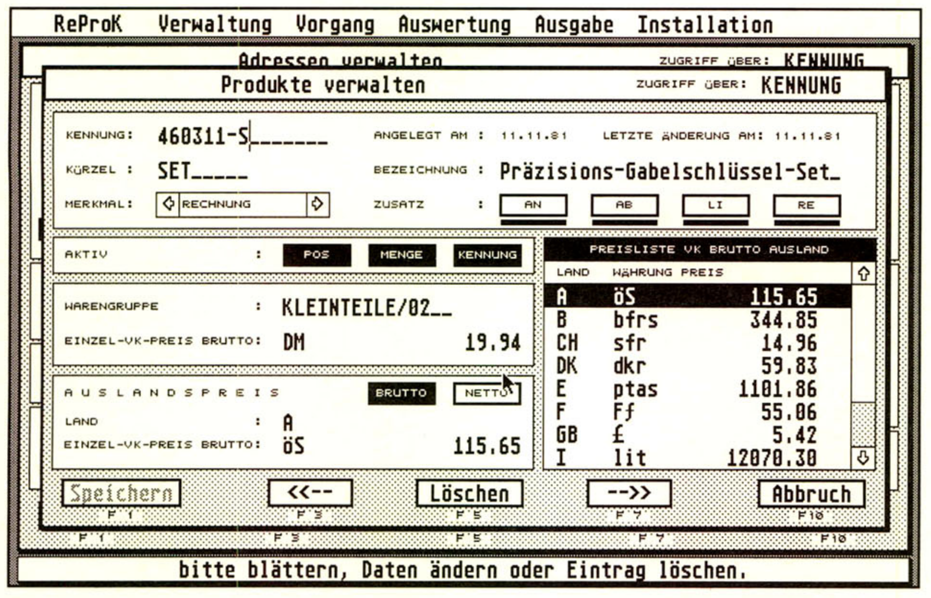
Durch viele verknüpfbare Auswahlkriterien können sogar die kniffligsten Listen ausgedruckt werden.



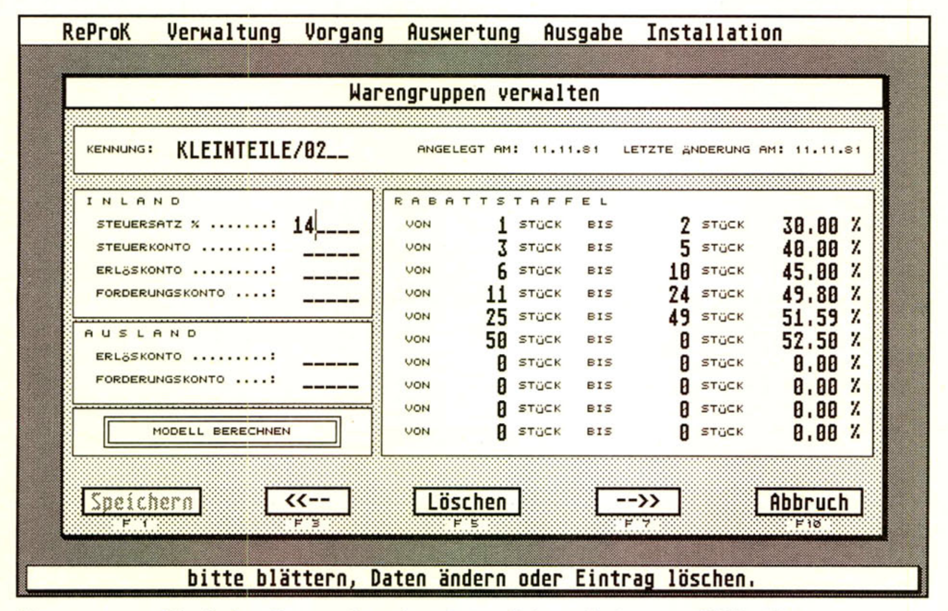
Die Suche nach bestimmten Adressen ist sehr einfach: Ein Zusatzfenster zeigt alle gefundenen Datensätze



Die Adressmaske: alle wichtigen Informationen auf einen Blick



Die Produktmaske: selbst in fremden Währungen ist der Preis sofort abrufbar



Warengruppen: hier finden übergeordnete Angaben zu Steuern, Rabatten und FiBu-Konto ihren Niederschlag.

RePro K Verwaltung Vorgang Auswertung Ausgabe Installation

länderspezifische Parameter einstellen

KENNZ.	NAME	WÄHRUNG	KURS	STEUER/ZOLL %	HEIMAT
A	Österreich	öS	6.9600	20.00	
B	Belgien	bfrs	20.5800	19.00	
CH	Schweiz	sfr	0.8200	9.30	
D	B. R. Deutschland	DM	1.0000	0.00	X
DK	Dänemark	dkr	3.6608	22.00	
E	Spanien	ptas	64.1000	16.00	
F	Frankreich	Ff	3.2750	18.60	
GB	Großbritannien	£	0.3128	15.00	
I	Italien	lit	714.2905	18.00	
NL	Niederlande	hfl	0.8975	20.00	

Speichern F1 Einfügen F4 Löschen F7 Abbruch F10

KENNZ.	NAME	WÄHRUNG	KURS	STEUER/ZOLL %	
D	B. R. Deutschland	DM	1.0000	0.00	HEIMAT

STAND: 14.05.89

bitte Daten ändern oder Eintrag löschen.

Ländermaske: Besonderheiten zu Parität und Steuer können für jedes Land getrennt eingestellt werden.

Die Masken

RePro K hat die Masken durchweg nach dem selben Schema aufgebaut, was ein einfaches Bedienen, Orientieren und Auffinden erleichtert. So befinden sich im oberen Bereich immer die Schlüsselfelder, über die gesucht werden kann. Die Adressmaske kennt deren vier: 1. die Kennung, 2. ein Kürzel, 3. Name-1 und 4. Name-2.

Hier ist der Hauptschlüssel, also jener, der für jeden Datensatz nur einmal existiert, die "Kennung". Das "Kürzel" kann als Unterschlüssel für verschiedene Adressen gleich definiert werden. Dies wäre beispielsweise bei verschiedenen Filialen einer Kette, oder Zugehörigkeiten mehrerer Adressen zu gleichen Organisationen wünschenswert. "Name-1" sollte üblicherweise die Firmenbezeichnung sein und hinter "Name-2" könnte sich der Ansprechpartner innerhalb dieser Firma verbergen.

Es erscheint mir wichtig, mindestens diese vier Unterscheidungsmerkmale in der Maske zu haben, denn was tun (sprach Zeus), wenn man zufällig die Adressennummer nicht auswendig weiß, aber der Name des Ansprechpartners bekannt ist? Viel mehr Suchschlüssel andererseits machen wenig Sinn, weil dies den Verwaltungsaufwand im Programm erhöht. Flexibel ist die Datensuche allemal. Es können Wildcards als Platzhalter verwendet werden: Der Stern "*" ersetzt dabei eine beliebige Zeichenfolge an beliebiger Position und das Fragezeichen "?" ein einzelnes Zeichen an bestimmter Position. So habe ich mit der Kombination "D??t*e*L*a*" = "Deutsche Lufthansa" und "Dritte Welt-Laden" gefunden. Oder probieren Sie mal einfach nur "*", dann erscheinen alle Datensätze (ist doch klar). Da diese Symbole "Gebrauchszeichen mit beschränkter Haftung (GmbH)" sind, sollte man sie selbst natürlich nicht innerhalb von Schlüsseln verwenden.

Dann gibt es noch die Eingabefelder für Anschrift, Telekommunikationsnummern, Bankverbindung, Länderzuordnung usw. Interessant ist die direkte Verbindung aus der Adreßmaske heraus zur Umsatzdatei und zum Posten "Offene Rechnungen". Da passiert es nicht mehr, daß ein Kunde fortwährend bestellt und alte Rechnungen noch offen sind (weil das Mahnwesen jemand anders bearbeitet als das Bestellwesen). Weiterhin kann direkt in ein Memofeld abgezweigt wer-

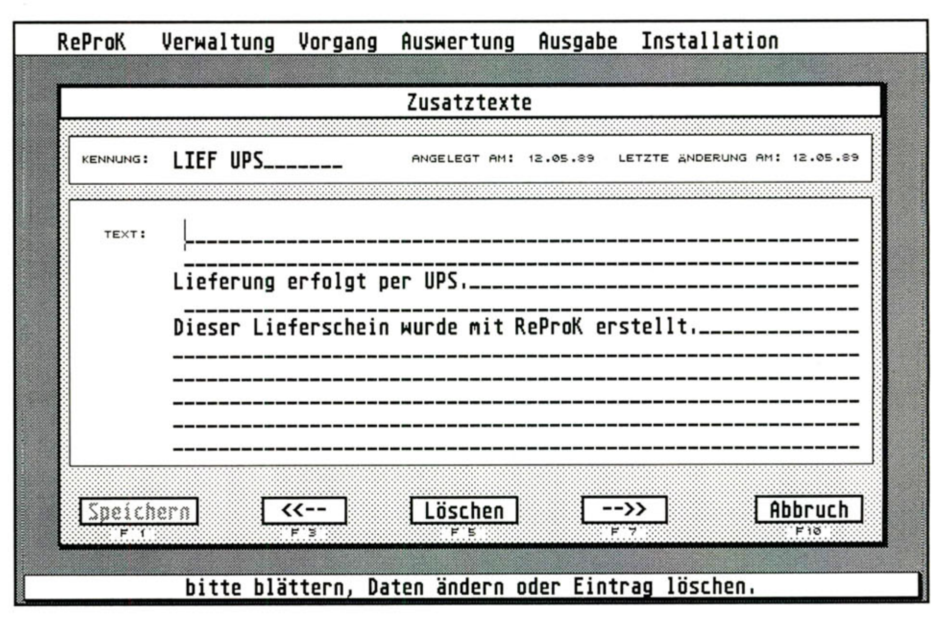
den. In diesem Notizblock ist Platz für wichtige Anmerkungen zur angewählten Adresse in Prosaform.

Noch ein globales Wort zu den Masken: Sie sind in zwei funktionale Bereiche aufgeteilt. 1. die Aufrufebene und 2. die Bearbeitungsebene. Man braucht es eigentlich nicht näher zu erklären, zuerst müssen alle wichtigen Komponenten zusammengesucht bzw. selektiert werden, dann folgen die weitergehenden Bearbeitungsschritte. Das Softwarehaus legte großen Wert auf die Tatsache, daß aus Gründen der Übersichtlichkeit auf ein und derselben Maske Daten neu angelegt, geändert und abgerufen werden können. So wird in der Aufrufebene (beispielsweise in der Adreßmaske) ein Schlüssel eingegeben, den das Programm kennt oder nicht kennt. Eine Infozeile gibt eine entsprechende Meldung. Kennt ReProK den Schlüssel, wird der Datensatz auf die Maske gebracht, andernfalls (wenn Schlüssel unbekannt) kommt eine leere Maske zur Neueingabe.

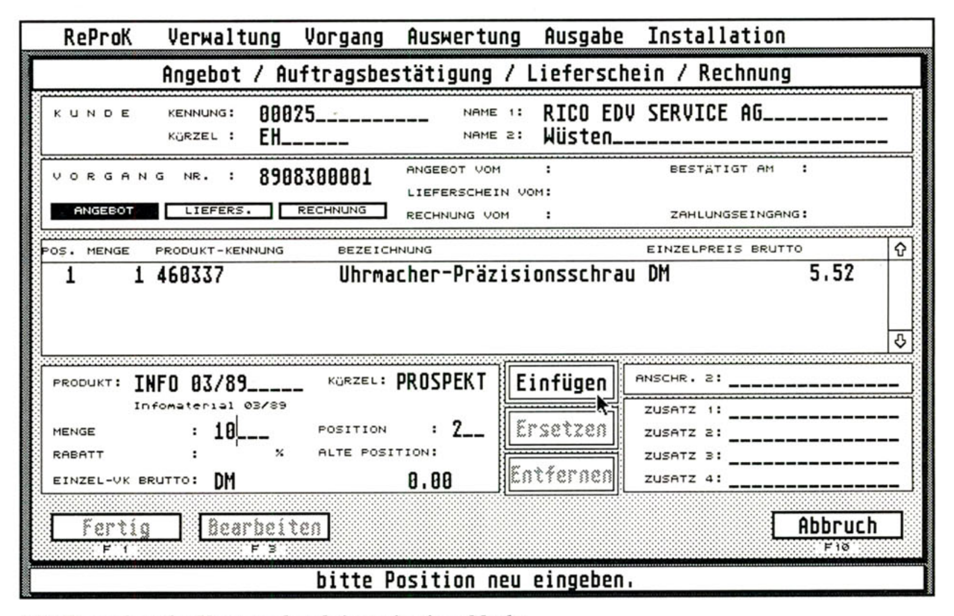
Produkte und Warengruppen

Zuerst habe ich es nicht verstanden, warum zwischen Produkten und Warengruppen unterschieden wird. Jetzt ist mir dieses ReProK-typische Vorgehen klar: "Produkt" kann sein, ein Handelsartikel oder eine Dienstleistung. "Warengruppen" sind dann meist mehrere ähnliche "Produkte" der gleichen Zielsetzung (was den Verkauf angeht). Die Warengruppe faßt quasi verschiedene Produkte unter einer Kathegorie zusammen. Alles was in der Produktmaske gespeichert wird, ist "einheitsbezogen", betrifft also immer nur ein Stück, oder eine andere Verpakkungseinheit. In der Maske für Warengruppen sind dann typische Angaben zu Steuersatz und Rabattstaffel enthalten. So ist die Warengruppe dem Produkt immer übergeordnet, so kann ein Produkt auch verschiedenen Rabattstaffeln unterliegen (beispielsweise für die sogenannte gemischte Abnahme). Das bringt in der Praxis erhebliche Arbeitserleichterung, da sich Preise oft für ähnliche Produkte zeitgleich ändern, oder Rabatte für diese gerne en block berichtigt werden sollen. So müssen für das einzelne Produkt die Daten zu Mehrwertsteuer und Rabattstaffel nicht mitgeschleppt werden (wie das normalerweise der Fall ist).

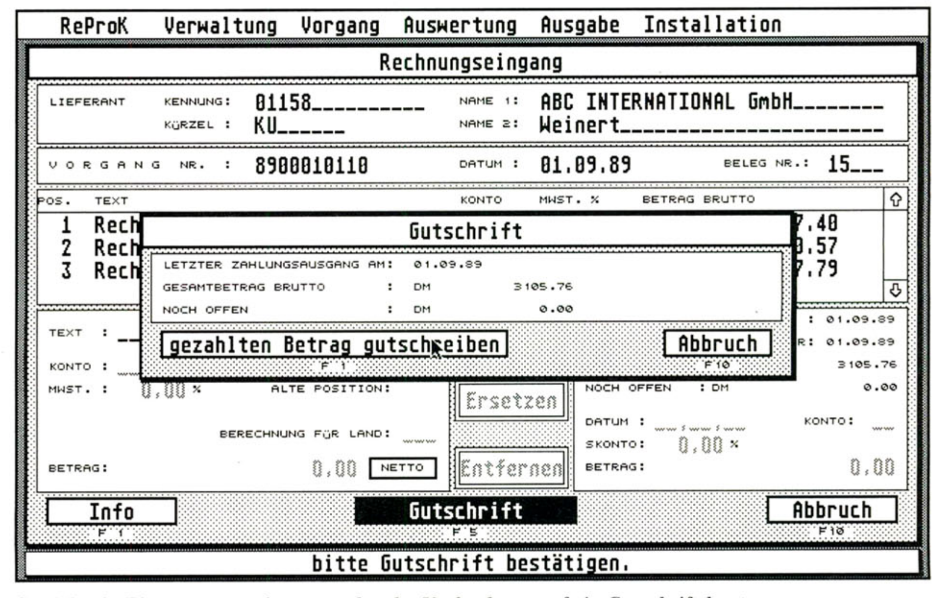
Sehr interessant ist in der Maske für Warengruppen die Anwahl "Modell be-



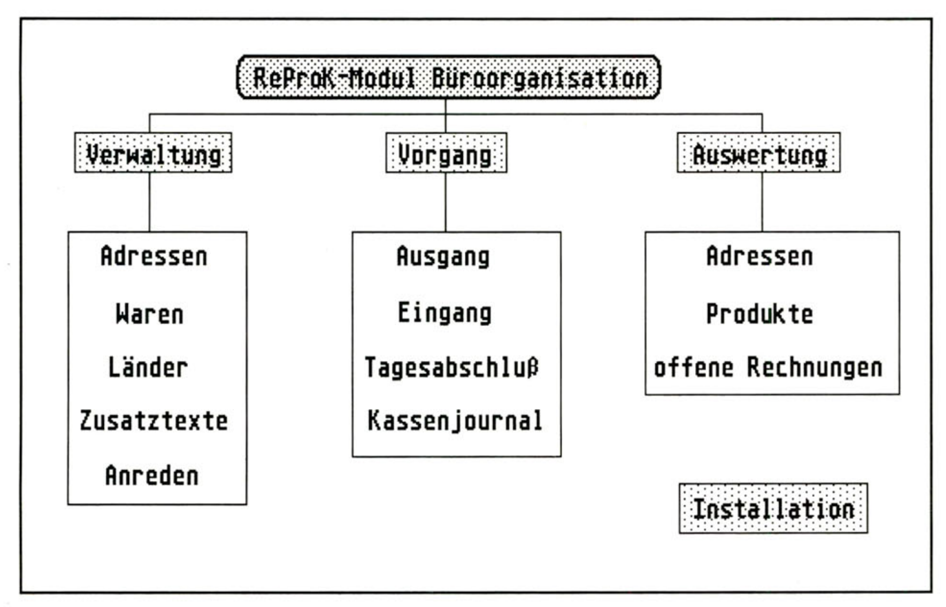
Individuelle Zusatztexte sind frei den Formularen zuordbar.



Alle Formulare der Vorgangsbearbeitung in einer Maske



So sieht ein Eingang aus, mit entsprechender Verbuchung auf ein Gutschriftskonto.



Der Stammbaum von ReProK

Verwaltung: Adressen, Produkte, Warengruppen / Länder, Zusatztexte, Anreden / Programmende

Vorgang: Ausgang, Eingang / Tagesabschluß / Kassenjournal / ✓ Rechnungsgeschäft, ✓ alternativ, Bargeschäft / Sofortverarbeitung, ✓ alternativ, ✓ Stapelverarbeitung / ✓ Vorgänge drucken

Auswertung: Adressen, Produkte / offene Rechnungen / Vorgänge löschen

Installation: Druckertreiber / Briefkopf / Angebote, Auftragsbestätig., Lieferscheine, Rechnungen / Listendruck, Etikettendruck / Buchhaltung, Rundschreiben / Nachkommastellen, ✓ akustische Warnung

Die Pull-Down-Menüs von ReProK im Überblick

rechnen". Das hilft, die Rabattierung optimal abzustufen. Es erfolgt daraus eine Tabelle, in der ausgehend von einem symbolischen Preis von 100 DM alles angezeigt wird, was für eine sinnvolle Rabattkalkulation wichtig ist: Stückzahl, Einzelpreis, Gesamtpreis für Stückzahl, prozentualer Unterschied zur voranstehenden Rabattstaffel. Da zeigen positive Werte die Höhe des Gewinns und negative die Höhe des (hoffentlich unerwünschten) Verlustes zur voranstehenden Rabattstaffel. Auch so etwas macht in der Praxis Sinn: Ladenhüter und (unsichere) neue Artikel müssen weit vorsichtiger rabattiert werden (also vielleicht mit ansteigender Verluststaffel) als Produktrenner. Daß auf diese Weise auch Mindermengenzuschläge als negative Rabatte möglich sind, sei nur am Rande erwähnt.

Diese Liebe fürs Detail zeigt, daß die Programmierer aus der Praxis kommen und mit diesem Programm schon effektiv gearbeitet wurde. Sinnvolle Funktionen stehen dort, wo sie gebraucht werden. Auch die Maske für Produktverwaltung hat Besonderes zu bieten: Wer hat es sich nicht schon öfters gewünscht, in Angebot, Auftragsbestätigung, Lieferschein oder Rechnung Erweiterungen, Erläuterungen oder Ergänzungen zur Produktbezeichnung anfügen zu können? Gerade im Angebot können ausführlichere (evtl. schmackhaftere) Beschreibungen zum Produkt kaufentscheidend sein. In ReProK sind solche Individualtexte frei anfügbar.

Der absolute Clou ist die besondere Auslandsverwaltung. Man weiß, daß es bestimmte Mentalitäten beispielsweise in unserem schönen Europa gibt, die gerne Angebote in Landeswährung sehen. Was macht es für einen positiven Eindruck, gleich ein Angebot für Österreich in Schillinge auszudrucken, wobei der Kunde nicht mehr mühsam die Parität umrechnen muß? Sind nicht gerade solche Kleinigkeiten im Geschäftsleben oft kaufentscheidend?

Zusatztexte

Gerade in Angeboten, Auftragsbestätigungen, Lieferscheinen und Rechnungen müssen oft unterschiedliche Texte voran- oder nachgestellt werden. Sprüche wie "Zahlung innerhalb 30 Tagen ohne Abzug" oder "Bei Zahlung bis 31.12.89 gewähren wir 3% Skonto" kennt fast jeder. Aber wenn ganz individuelle Texte nötig werden? Da die Zusatztexte auch einen Schlüssel erhalten und unabhängig vom Formular gespeichert werden, ist eine regelrechte Bibliothek solcher Sprüche machbar. Oder wollen Sie jedes Jahr aufs neue diesen Text neu eintippen: "Unseren Geschäftskunden ein frohes Weihnachtsfest ..."?

Der Vorgang

Jeder Geschäftsvorfall wird sofort als Vorgang festgehalten. Jede Anfrage von Interessenten oder jeder Rechnungseingang von Lieferanten ist ein Vorgang für sich. Grundsätzlich unterschieden wird nach Ausgang und Eingang, denn jedes Geschäft läuft ja auf diesen zwei Schienen. Jeder Vorgang erhält eine Nummer (Muster: YYMMDDXXXX = Jahr, Monat, Tag, lfd. Nummer) und läuft dann eine regelrechte Hierarchie durch: Angebot, Auftragsbestätigung, Lieferschein, Rechnung, Zahlungseingang, Gutschrift, Gutschriftsbuchung, und das alles in einer einzigen Maske!

Selbst wenn Vorgänge abgewickelt sind, bleiben sie gespeichert - weil man sicher noch gelegentlich dort nachsehen muß. Gleich zu Anfang wird auch jeder Vorgang mit einer Adresse verknüpft. Es ist wirklich angenehm, fast alles mit der Maus zusammenstellen zu können - nur für die Mengeneingabe braucht man noch die Tastatur.

Erinnern Sie sich noch an die Rabattgestaltung? 1. Kundenspezifischer Grundrabatt in der Adreßmaske, 2. mengenabhängiger Rabatt in der Warengruppe und 3. hier: produktbezogener Rabatt bei jedem Einzelposten des Vorganges.

Zur Vorgangsbearbeitung: Der Kunde ist per Adreßkennung gewählt und bekommt sofort eine neue Vorgangsnummer zugeordnet. Es kann natürlich auch ein laufender Vorgang weiterbearbeitet werden. Anschließend folgt die Auswahl der einzelnen Vorgangspositionen (Produkte) mit der Mengenangabe. In der Mitte des Vorgangsmenüs erscheinen dann die einzelnen Positionen untereinander. Durch "Fertig" wird der Vorgang als (vorerst) abgeschlossen behandelt und gespeichert.

Jetzt erscheint das Angebotsformular auf dem Bildschirm (wer kennt "WYSIWYG" nicht? = was du siehst, ist, was du kriegst) und alle Positionen und die Zusammenrechnung können geprüft werden. Auch Rückkehr in den Vorgang, problemloses Ändern der Positionen sowie Neuberechnung (sogar unter Änderung von Rabatt und Preis) sind jederzeit möglich. ReProK nennt diesen Zustand der Vorgänge "Sammlung", erst wenn sie gedruckt sind, nehmen sie den Zustand "nicht löschbar" ein, sind aber dennoch "stornierbar".

Tagesabschluß und Kassenjournal: Entweder per Pull-Down-Menü im Programm oder bei Programmende ausführbar ist der Tagesabschluß. Jetzt werden alle gesammelten Vorgänge ausgedruckt. Auf Wunsch kann man jetzt auch die Bargeschäfte per Kassenjournal drucken.

Auswertung

Über eine vielfältig gestaltete Maske sind verschiedene Auswertungen möglich. So können Adressen oder Produkte in Listenform ausgedruckt werden. Die Kriterien für die Zusammenstellung der Daten sind so variabel filterbar, daß kaum ein noch so undenkbarer Fall übrig bleibt: Ob Interessent, Kunde, Lieferant, nach bestimmten Kürzeln, Ländern, Postleitzahl, Datum früher/später als, Umsatz mindestens/höchstens, Anlagezeitraum für Vorgänge von/bis, Produkten oder Notizblockvermerken.

Installation

Nicht zu verwechseln mit dem Startprogramm, das ReProK von der Originaldiskette auspackt, verbirgt sich hinter diesem Menüpunkt eine umfangreiche Treiberinstallation. Für jedes Formular sind Anpassungen vorgesehen, mit Attributen des Schriftbildes zu jeder Position, Formaten des Papiers, Angaben zu Zusatztexten usw. In der Regel braucht man diese Einstellungen nur einmal vornehmen, es zeigt aber auch hier, wie flexibel Änderungen (bei neuen Druckern) durchführbar sind.

Sie haben drei Drucker in der Firma? Laser für Briefe, Angebote u.ä., 24-Nadel für Etiketten und 9-Nadel für Lieferschein, Rechnung und Listen? Die können Sie jetzt alle an ReProK anschließen - oder reichen die Schnittstellen nicht aus?

Geschwindigkeit ist keine Hexerei

Ich hab ReProK einmal auf die Probe gestellt: (Konfiguration: MEGA ST4, SH 205, Blitter-TOS 1.2)

1. Alle 2500 Adressen (nein - ich kenn die Leute nicht alle) anzeigen (mit "*" als Kennung) = 2 min 30 sec.
2. Alle Adressen suchen mit einer "1" in Kennung ("*1*") = 1 min 35 sec (1447 Datensätze)
3. Alle Adressen suchen mit einer "11" in Kennung ("*11*") = 20 sec (142 Datensätze)
4. Alle Adressen suchen mit einer "111" in Kennung ("*111*") = 8 sec (12 Datensätze)

Man kann es ohne Übertreibung sagen: ReProK ist extrem schnell!

Für Programmierer

ReProK ist vollständig in C geschrieben und besteht aus dem Applikationscode, einer Environment-Library und der Datenbank selbst. Der Datenbankfunktion liegt der "c-tree"-ISAM-Filemanager zugrunde, welcher grundsätzlich schneller arbeitet als Relationale Datenbanken. Die Verbindungen der einzelnen Dateien sind nicht in der Datenbank selbst festgehalten, sondern in der Programmlogik kodiert worden. Da der Firma STAGE MICROSYSTEMS "c-tree" im C-Source code zur Verfügung steht, sind Portierungen auf andere Rechner sofort möglich.

Für Ihre Buchhaltung

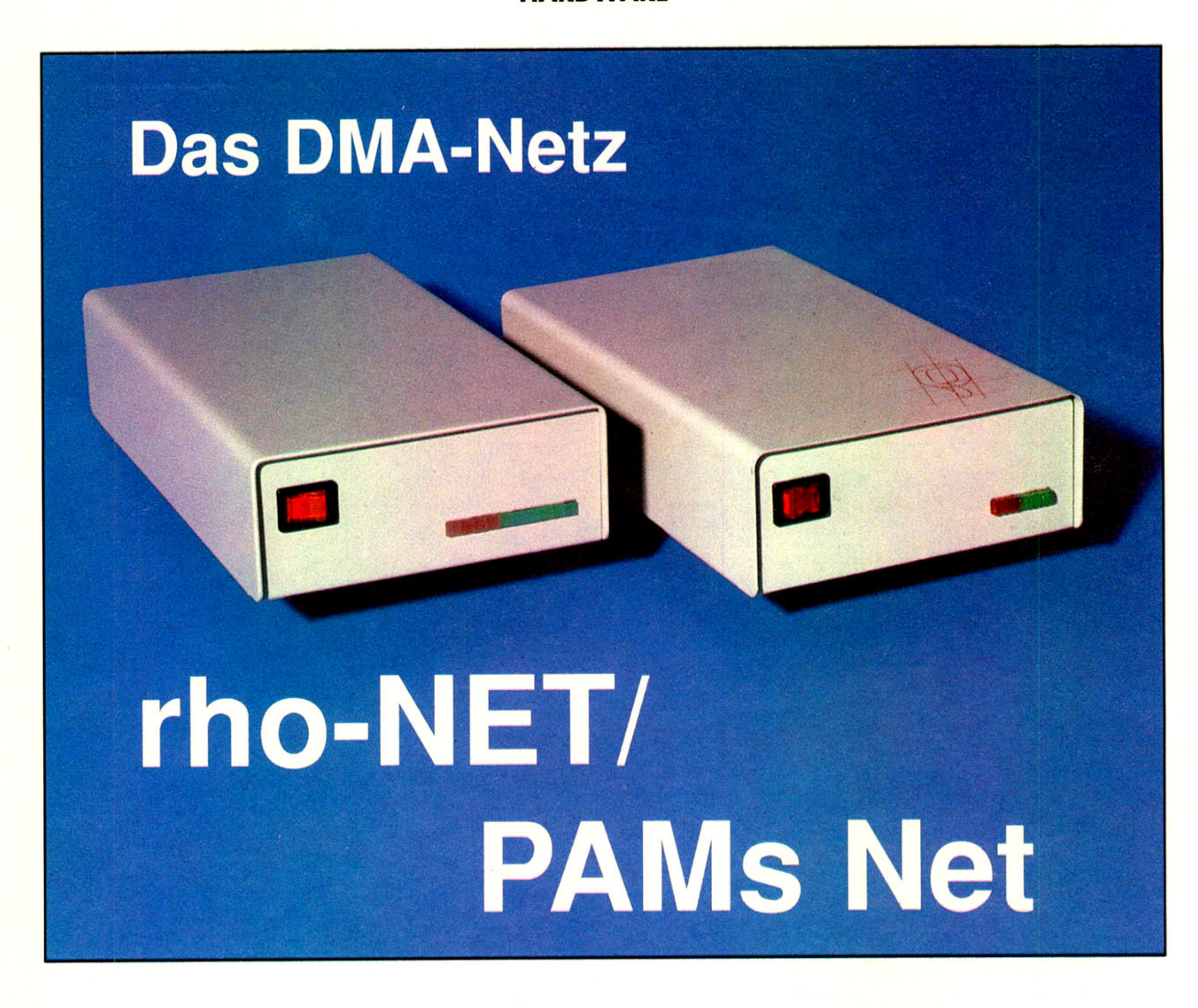
Durch eine beispielhafte Übereinkunft mit der Firma C.A.S.H. GmbH, Augsburg und STAGE MICROSYSTEMS (Hersteller von ReProK) ist eine Softwareschnittstelle zur T.I.M.-Finanzbuchhaltung eingebaut worden. T.I.M. übernimmt alle fakturierbaren Vorfälle. Gerade über diese Kooperationsbereitschaft war man bei STAGE MICROSYSTEMS angenehm überrascht - ein Beispiel, das Schule machen sollte.

Fazit

ReProK besticht durch seine Geschwindigkeit beim Suchen und Sortieren. Wer sich an das Hantieren mit der Maus gewöhnt hat, kann auch in den Menüs ganz schön flinke Arbeit verrichten. Einige Funktionen, an die ein "normaler" Programmierer vielleicht nie gedacht hätte zeigen, daß Funktionalität und Überlegung dahinter stecken. Es gibt für den Interessierten eigentlich nur eins: ReProK beim autorisierten Fachhändler (ein bißchen Zeit mitbringen) ansehen und ausprobieren. Oder sich eine DEMO mit Handbuch (!) für 39 DM kommen lassen. ReProK kostet für ATARI-ST: 598,- DM; für MS-DOS: 798,- DM. ReProK wird nur über ausgesuchte Händler abgegeben, außerdem ist eine kostenlose Telefon-Hotline eingerichtet. Wenn individuelle Anpassungen von Masken und Formularen (z.B. bei Hinzunahme von neuen Datenfeldern u.ä.) oder Anwenderschulung für Firmen gewünscht sind, werden faire, marktgerechte Preise zugesichert.

Hersteller:
STAGE MICROSYSTEMS, Velbert

Bezugsquelle:
Hans Richter
Hagener Straße 65
5820 Gevelsberg
Telefon: 02332/2706



Das DMA-Netz rho-NET/PAMs Net

Einer der Anbieter für Netzwerke ist die Firma Rhothron aus Homburg mit PAMs Net. Was das Netzwerk, das über den DMA-Port angeschlossen wird, leistet, lesen Sie in diesem Artikel.

Um das rho-NET/PAMs Net zu betreiben, wird jeder ATARI-Rechner an einen Netzknoten angeschlossen (Titelbild). Die Verbindung zwischen den Geräten wird über den DMA-Port hergestellt. Der Port ist natürlich durchgeschleift, so daß weiterhin auch die üblichen Geräte wie Festplatten und Laserdrucker betrieben werden können. Die Verbindung zwischen den einzelnen Netzknoten ("Nodes") wird über Koaxialkabel hergestellt. So lassen sich auch große Entfernungen zwischen den Nodes ohne Probleme herstellen. Wenn mehr als zwei Netzknoten angeschlossen werden sollen, werden alle Geräte hintereinander verkabelt, also Gerät 1 mit Gerät 2, Gerät 2 mit Gerät 3 etc. Freibleibende Enden (beim ersten und letzten Gerät der Reihe) werden dabei mit einem 50 Ohm-Widerstand abgeschlossen. Wer bereits mehrere Festplatten und/oder Laserdrucker in Betrieb hat, muß sich keine Sorgen um die Geräteadressen der Knoten machen: Sie läßt sich über DIP-Schalter komfortabel einstellen. Dem Netz ist es gleichgültig, welche Adressen die Knoten besitzen; lediglich zwei gleiche Nummern dürfen nicht vorkommen.

Im Hintergrund

Als Anwender kann man das Netz praktisch nur beim Bootvorgang bemerken. Hier erscheint eine kurze Copyright-Meldung, und man kann sich unter seinem Benutzernamen mit einem vorher festgelegten Paßwort einloggen, also dem System klarmachen, wer man ist. Dieses Paßwort kann der Benutzer jederzeit ändern. Es wird verschlüsselt abgespeichert und läßt sich so nicht mehr ohne weiteres dechiffrieren. Jedem Benutzer kann ein eigenes Zugriffsrecht gegeben werden, das beim Booten aus der Konfigurationsdatei eingelesen wird. Diese Datei ist eine ASCII-Datei und kann leicht abgeändert werden. Vorteilhaft dabei ist, daß man quasi jedem Benutzer des Netzes sein eigenes Desktop mit nur für ihn geltenden Zugriffsrechten geben kann. So können beispielsweise auch Gäste zugelassen werden, die zwar von Laufwerk C alle Programme laden, jedoch weder kopieren, noch löschen dürfen. Anders als bei anderen Netzwerkprogrammen muß hier nicht alles über ein bestimmtes Laufwerk (beispielsweise "N") geroutet werden, sondern man kann so viele Laufwerke anmelden, wie es das GEMDOS zuläßt: So kann man beispielsweise das Laufwerk C von Platz 3 als Laufwerk G von Platz 6 anmelden. Der Vorteil dieser Handhabe liegt auf der Hand: Es existiert praktisch kein Programm, welches sich nicht mit dem Netzwerk verträgt, da übliche Dateinamen verwendet werden. So werden quasi alle Programme netzwerktauglich! Das Desktop eines Superusers mit allen verfügbaren Laufwerken von 2 Plätzen sehen Sie in Bild 1. Doch beim Booten zeigt sich ein weiterer großer Vorteil des Netzwerks, den andere Programme vermissen: Jeder Rechner kann über ein anderes System

gebootet werden. Zum Startvorgang ist also keine Diskette nötig, sondern Platz 2 kann beispielsweise mit der Festplatte oder der Diskette von Platz 4 booten.

TOS-like

Wenn man ein Laufwerk, egal ob ein "eigenes" oder das Laufwerk eines anderen Platzes, öffnet, merkt man keinen Unterschied mehr zum normalen Arbeiten mit dem TOS, da von nun an alles über die intelligenten Netzknoten verwaltet wird. Auch ein Geschwindigkeitsnachteil ist nicht festzustellen, sofern nicht 2 Plätze gleichzeitig auf eine Datei zugreifen wollen: Da der gesamte Zugriff über den DMA-Port läuft, werden die Daten praktisch ungebremst an ihren Empfangsort weitergeleitet. Programme können hier nicht nur geladen, sondern auch kopiert oder angesehen werden - eben mit den normalen TOS-Funktionen. Dabei ist es, sofern man die Zugriffsrechte dafür besitzt, natürlich auch möglich, eine Datei von Platz 3 nach Platz 4 zu kopieren, selbst wenn man sich selbst auf Platz 2 befindet. Da natürlich in einem Netz mehrere Benutzer existieren, müssen die Zugriffsrechte verteilt werden. Hier kann man unterscheiden zwischen den möglichen Rechten Lesen, Schreiben oder Lesen/Schreiben. Hier kann weiterhin je nach Laufwerk unterschieden werden, so daß man beispielsweise auf Laufwerk C nur lesen, auf Laufwerk D nur schreiben darf etc. Nicht nur ganze Laufwerke, sondern auch einzelne Directories kann man mit diesem Schutz versehen, so daß man sehr variabel die Dateien auswählen kann, die jeder Benutzer lesen bzw. schreiben darf. Natürlich existieren noch andere Geräte als Festplatten in der ATARI-Welt. So kann auch an einen der Netzrechner der ATARI-Laserdrucker SLM 804 angeschlossen werden. Auf diesen kann dann von jedem Netzknoten aus zugegriffen werden. Vorteilhaft dabei ist, daß der normale ATARI-Lasertreiber verwendet werden kann. Alle Daten, die an den Laserdrucker gehen sollen, werden abgefangen und in eine Spooldatei geschrieben. So werden alle Texte nacheinander ausgedruckt und Wartezeiten durch den langsamen Drucker haben ein Ende. Leider funktioniert im Netz das *SETUP630.ACC* nicht und man kann keinen Seitenvorschub durchführen. Hier kann man sich jedoch abhelfen, indem man eine Datei anlegt, in der lediglich ein Formfeed enthalten ist, und diese ausdruckt: Das Ergebnis bleibt das gleiche. Der Laserdrucker kann jedoch nicht nur mit Textausdrucken beschäftigt werden, sondern auch mit Hardcopies. Bis die Hardcopy den ATARI-Laser erreicht hat, vergeht höchstens eine Sekunde. Nicht nur der Laserdrucker kann dezentral betrieben werden, sondern auch Streamer und Modems (!): Ein Modem, das an Einheit Nummer 3 hängt, kann problemlos von einem anderen Knotenpunkt aus benutzt werden. Rechner 3 wird dadurch nicht im geringsten belastet, sondern kann einfach weiterbenutzt werden. Da das rho-NET/PAMs Net transparent arbeitet, werden Fehler durch normale TOS-Meldungen angezeigt. Das ergibt zwar nicht immer sinnvolle Fehlermeldungen, ist jedoch vollkommen ausreichend.

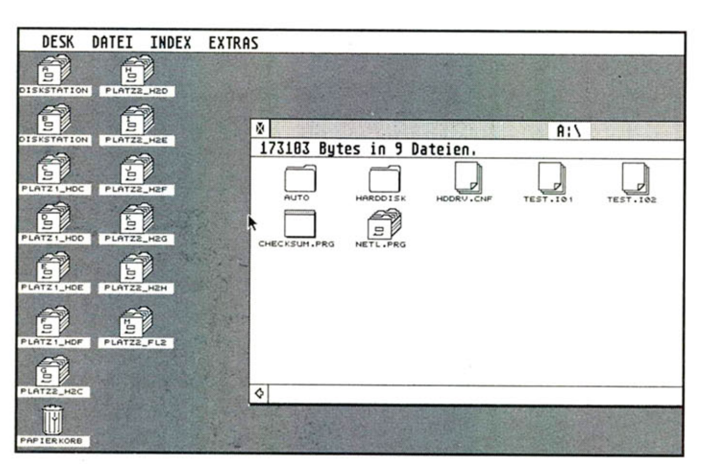


Bild 1: Das Desktop eines Superusers mit allen verfügbaren Laufwerken von 2 Plätzen

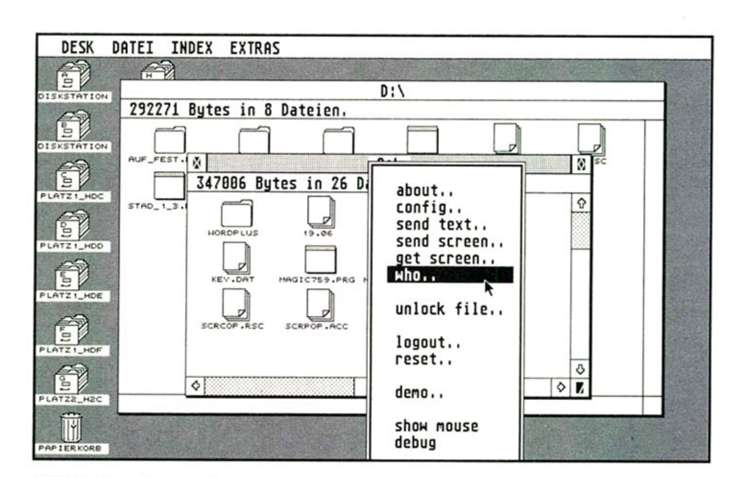


Bild 2: Das Popup-Accessory

Popup-Accessory

Außer den erweiterten TOS-Funktionen steht natürlich noch ein weiteres Hilfsmittel zur Verfügung: das Popup-Accessory (Bild 2). Das Accessory stellt eine reine Serviceleistung dar und kann jederzeit, auch in laufenden Programmen, durch die rechte Maustaste aufgerufen werden. Hier können beispielsweise auf einfache Art und Weise Nachrichten an andere Netzteilnehmer geschickt werden (Bild 3). Der Empfänger wird sofort vom Eintreffen einer Nachricht informiert. Doch nicht nur an einzelne Teilnehmer kann eine Nachricht verschickt werden, auch ein "Rundschreiben" an alle angeschlossenen Stationen kann verschickt werden. Interessanterweise kann man auch von einem Knotenpunkt die gesamte Steuerung eines anderen Knotenpunkts übernehmen. Dabei wird jeweils der gesamte Bildschirminhalt der zu steuernden Einheit zum steuernden Rechner geschickt und der steuernde Rechner verhält sich einschränkungslos wie der gesteuerte ATARI. Dieser Programmteil hat den Namen *Bagdad* bekommen und ist angelehnt an das Mac-Programm Timbuktu. *Bagdad* übermittelt 7 Bilder pro Sekunde. Dadurch wird die "Fernsteuerung" quasi zum Kinderspiel. Einige Programme lassen nach ihrem Start die Maus verschwinden. Dieses lästige Problem ist durch das Popup-Accessory endgültig behoben, da man jederzeit den Mauszeiger wieder einstellen kann. Möchte man, beispielsweise zur Mittagspause, seinen Platz kurzzeitig verlassen, kann man sich auch wieder "ausloggen". Dadurch ist gewährleistet, daß niemand sich unbefugt an seinen Daten zu schaffen machen kann, da man sich dazu erst wieder einloggen muß: Dazu sind jedoch eine eigene Benutzerkennung und ein Paßwort vonnöten. Als nützliche Dreingabe existiert auch ein kleiner Speichermonitor, der hexadezimal und dezimal die Aktivitäten der anderen Netzknoten anzeigt. In der rechten unteren Ecke wird dabei die prozentuale Auslastung des eigenen 68k-Prozessors angezeigt. Dieses kleine Utility kann beispielsweise

sehr nützlich für Programmierer oder zur Ferndiagnose bei einem Fehler sein.

rho-NET für Programmierer

Für Programmierer besteht die Möglichkeit, das Netzwerk in eigene Programme einzubinden. Für diesen Zweck ist die benötigte Headerdatei schon bei der Programmlieferung enthalten. Das rho-NET/PAMs Net richtet sich nach dem sogenannten CSMA/CD bzw. -/CA ("Carrier Sense Multiple Access with Collision Destection" / -"Collosion Avoidance") -Standard. Die an das Bussystem angeschlossenen Knoten "hören" quasi alle Anforderungen mit und wissen so sofort, wann gesendet bzw. empfangen werden muß. Die Übertragung selbst läuft nach dem SDLC-Protokoll. Wird ein Datenpaket von einem Netzknoten zu einem losgeschickt, kann der empfangende Knoten das Paket zwischenspeichern: Insgesamt können bis zu 100 Pakete mit einer Gesamtlänge von 20 kB gepuffert werden. Das erhöht den Datendurchsatz erheblich. Auch während ein Netzknoten Daten sendet, registriert er weiterhin alle Aktivitäten im Netz und kann sich so unter Umständen zurückziehen, wenn er dazu aufgefordert wird. Das Netz ist angelehnt an das Ethernet (nach IEEE-Norm 803.2) und erfüllt alle bei der ATARI-Netzwerkkonferenz festgelegten Anforderungen. Dem Programmierer stehen neue GEMDOS-Funktionsaufrufe zur Verfügung, die die Nummern 96 bis 127 besitzen. Dadurch ist es möglich, das Netzwerk unter jeder Programmiersprache ohne Probleme zu nutzen. Die Übertragungsrate ist mit 2 MBit/s angegeben. Der Datendurchsatz ist natürlich geringer und beträgt ungefähr 150 bis 200 kB/s (wenn von RAM zu RAM übertragen wird) bzw. zirka 100 kB (wenn auf Dateien zugegriffen werden muß). Damit ist das rho-NET/PAMs Net immerhin schneller als DECNet auf zwei VAXen, bei denen der Datendurchsatz lediglich zirka 60 kB/s beträgt. Die Netzknoten haben grundsätzlich gleiche Zugriffsrechte, unabhängig von der Stellung im Netz (Server, Gast etc.). Das wird durch das "Listen before and while talking" möglich. Wie bereits beschrieben, wird jede Aktivität registriert und die Netzknoten sprechen nur dann an, wenn die anliegenden Daten für sie bestimmt sind. Dadurch ist es auch möglich, einen defekten oder ausgeschalteten Knoten im Netz zu haben, da er zwar nicht mehr "mithört", dadurch aber auch nicht die Funktion der anderen Netzknoten stört.

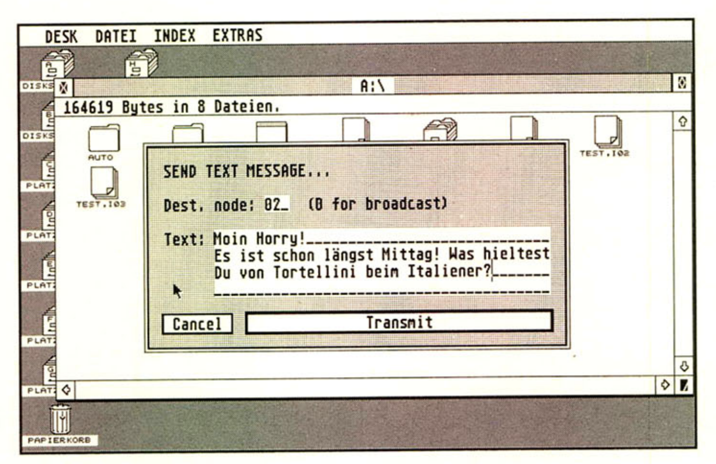


Bild 3: Es lassen sich ganz einfach Nachrichten an andere Netzteilnehmer verschicken.

Recordlocking

Normalerweise dürfen nicht zwei Programme gleichzeitig lesend und schreibend auf eine Datei zugreifen, da man seine Daten sonst gleich einem Reißwolf anvertrauen könnte. Daher ist es nur zulässig, wenn zwei Knoten gleichzeitig eine Datei lesen. Alle anderen möglichen Kombinationen weist das rho-NET/PAMs Net mit einer Fehlermeldung entrüstet ab. Nun ist das Sperren einer gesamten Datei nicht unbedingt sinnvoll, sondern in bestimmten Fällen eher sinnlos: Man denke an eine Adimens-Datei von über 1 MB länge. Hier ist es möglich, das "Record-Locking" anzuwenden, bei dem immer nur der Teil einer Datei gegen einen weiteren Zugriff geschützt wird, der gerade beschrieben wird. Dadurch wird es beispielweise möglich, daß sich mehrere Anwender in einer Datei zu deren Bearbeitung tummeln können, ohne sich gegenseitig zu behindern. Natürlich können auch Programmierer auf diese nützlichen Funktionen zurückgreifen.

Die Preise

Das rho-NET/PAMs Net kostet, je nach Ausstattung, unterschiedlich viel. Ein Netzknoten inklusive Software schlägt mit DM 1498,- zu Buche. Damit beträgt der günstigste Preis (für 2 Rechner) DM 2996,-. Dazu benötigt man noch Koaxialkabel mit BNC-Steckern sowie Abschlußwiderstände, die mit jeweils 45,60 DM (10 Meter) bzw. 39,90 DM bezahlt werden wollen. Für den MegaST existiert eine Steckkarte für DM 1698,- inklusive Software. Bei dieser Steckkarte besteht der Unterschied, daß mit einer Übetragungsgeschwindigkeit von ungefähr 10 MBit/s gearbeitet wird. Außerdem ist richtet sich diese Karte 100%ig nach dem Ethernet-Standard. Weiterhin kann man Lizenzen für VAXen/VMS (zum transparenten Zugriff auf VAXen), TOS-Shells (um mit dem TCP-IP-Protokoll arbeiten zu können) und weitere Adapterarten erwerben. Das Rhothron-Netz / PAMs Net läuft unter TOS 1.0, 1.2 und 1.4, eine Anpassung an TOS 1.6 (ATARI ST[E]) wird wahrscheinlich bereits fertig sein, wenn dieser Artikel erscheint.

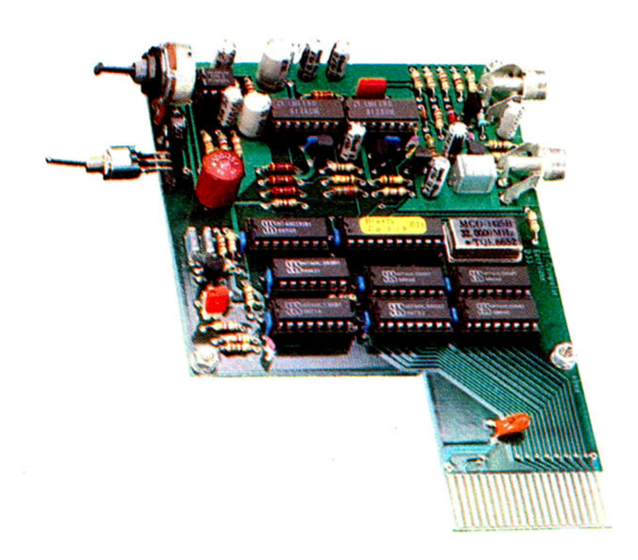
Fazit

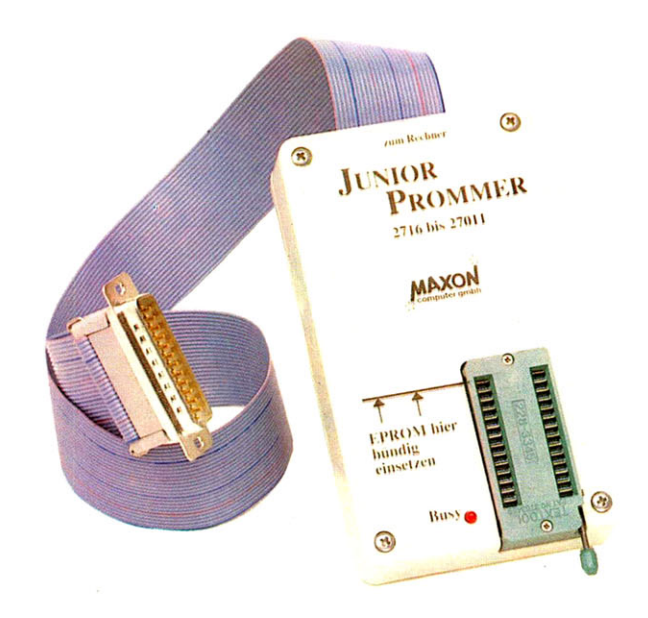
Wer ein Netz benötigt, das mit Sicherheit kompatibel zu fast allen Programmen ist, welches überaus schnelle Datentransferraten aufweisen kann, gute Geräte-Share-Möglichkeiten bietet und zudem relativ einfach zu bedienen und zu intallieren ist, ist mit dem rho-NET/PAMs Net sehr gut beraten. Dadurch daß viele Geräte dezentralisiert angesteuert werden können und nicht neue Dateipfade hinzugefügt werden, kann das Netz universell eingesetzt werden.

MP

Rhothron GmbH
Entenmühlenstraße 57
6650 Homburg/Saar
Tel. 06841/64067

PAM Software
Carl-Zuckmayer-Straße 27
6500 Mainz-Drais
Tel. 06131/476312 ab 14.30h





Easytizer - der Videodigitizer ohne Geheimnisse

Mit dem Easytizer können Sie beliebige Videosignale von einer Schwarzweiß- oder Farb-Kamera, Videorecorder oder direkt vom Fernsehgerät (mit Composite Video Ausgang) digitalisieren und somit auf dem Bildschirm Ihres ATARI ST sichtbar machen.
Der Easytizer wird am Modul-Port des ST angeschlossen. Die Auflösung beträgt 800x600 Bildpunkte, so daß in mittlerer Auflösung 640x200 Bildpunkte in vier Graustufen dargestellt werden können. In dieser Betriebsart werden 12,5 Bilder in der Sekunde wiedergegeben. Im hochaufgelösten Modus werden 640x400 Bildpunkte in Schwarzweiß wiedergegeben.

Besondere Merkmale des Easytizers:

- Software vollständig in Assembler
- Abspeichern der Bilder im DEGAS-Format
- Von STAD und Sympatic Paint ansteuerbar
- Ein beliebiger Bildausschnitt kann in ein mit der Maus wählbares Format vergrößert und verkleinert werden
- Spiegeln eines Bildes in horizontaler und vertikaler Ebene
- Animation mit beliebig vielen Bildern möglich, nur durch die Kapazität des Rechners begrenzt (beim Mega ST4 über 100 Bilder)
- Eingebauter Druckertreiber für die mittlere Auflösung für NEC P6/P7 und EPSON oder Kompatible
- Wahlweise automatische oder manuelle Helligkeitseinstellung
- Schnappschuß

Lieferumfang:

1) Fertiggerät
   komplett aufgebaut und geprüft, inclusive Diskette mit der Easytizer Software und Bedienungsanleitung
2) Teilsatz
   Für Bastler liefern wir einen Teilsatz bestehend aus:
- Doppelseitiger, elektronisch geprüfter Platine mit Lötstoplack und Bestückungskungsaufdruck sowie vergoldeten Anschlußkontakten
- fertig programmiertes GAL 16V8
- Quarzoszillatormodul 32 MHz
- Diskette undBedienungsanleitung.

Klein, kompakt und leistungsstark der *Junior Prommer*

Der Junior Prommer programmiert alle gängigen EPROM-Typen, angefangen vom 2716 (2 KByte) bis zum modernen 27011 (1 MBit). Aber nicht nur EPROMs, sondern auch einige ROM- und EEPROM-Typen lassen sich lesen bzw. programmieren.
Zum Betrieb benötigt der Junior Prommer nur +5 Volt, die am Joystick-Port Ihres ATARI ST abgenommen werden, alle anderen Spannungen erzeugt die Elektronik des Junior Prommers. Die sehr komfortable Software, natürlich voll GEM unterstützt, erlaubt alle nur denkbaren Manipulationen.
Selbstverständlich läßt sich ein 16-Bit Word in ein High- und Low-Byte zerlegen. Fünf Programmieralgorithmen sorgen bei jedem EPROM-Typ für hohe Datensicherheit. Im eingebauten Hex/ASCII-Monitor läßt sich der Inhalt eines EPROMs blitzschnell durchsuchen oder auch ändern.

Alles dabei!

Bemerkenswert ist der Lieferumfang, so wird z.B. das Fertiggerät komplett aufgebaut und geprüft im Gehäuse mit allen Kabeln anschlußfertig geliefert. Auf der Diskette mit der Treibersoftware befinden sich noch RAM-Disk und ein Programm zum Erstellen von EPROM-Karten, ferner wird der Source-Code für Lese- bzw. Programmierroutinen mitgeliefert und last but not least ist im Bedienungshandbuch der Schaltplan abgedruckt.

Typ

INT. ID.			
EPROM'S			
2716	25V		
2732	25V	2732A	21V
2764	21V	2764A	12V
27128	21V	27128A	12V
27256	12V	27256	21V
27512	12V	27513	12V
27011	12V		
ROM'S			
4732		4764	
47128		47256	
EEPROM'S			
X2804A		X2816A	
X2864A		X28256A	

Übersicht der mit dem Junior Prommer programmierbaren Speichertypen

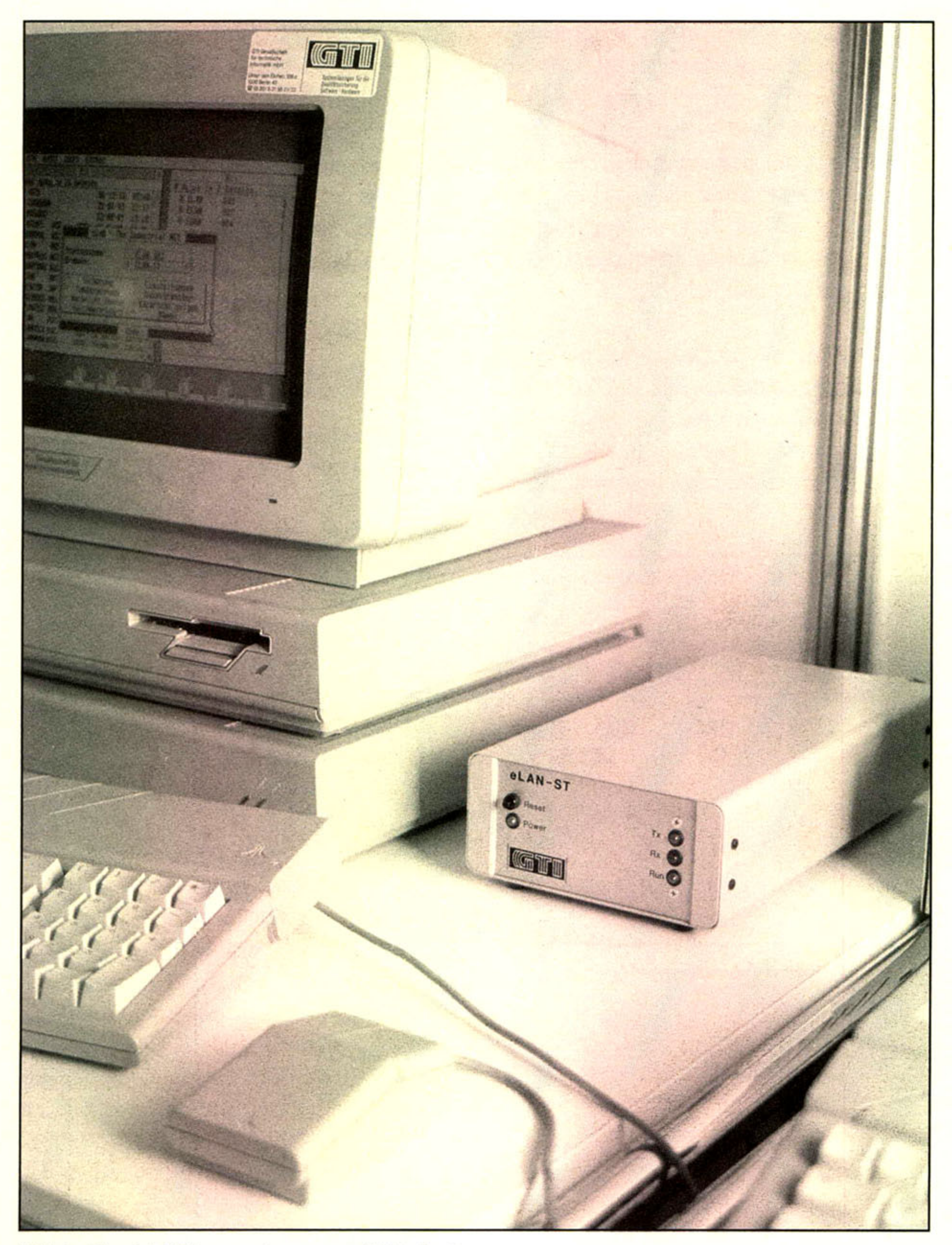


Bild 1: Ein eLAN-Knoten als externes DMA-Gerät

Mit eLAN ans industrielle Netz

eLAN ist ein Netzwerk, das sich mit großer Fehlersicherheit besonders für industrielle Anwendungen empfiehlt. Die Eckdaten sind eine Übertragungsrate von 1 MBit/s, bis zu 254 anschließbare Stationen und eine Vernetzung in einer Busstruktur mit Token-Passing.

Zum Betrieb von eLAN wird jeder ATARI mit einem Netzknoten ausgerüstet, der die Intelligenz zum Netzbetrieb enthält. Es gibt die Knoten in zwei Ausführungen, je nach verwendetem Rechner: für ältere STs als externes DMA-Gerät (Bild 1) und für MEGAs als Steckkarte für den Prozessorbus. Nach Angaben der Hersteller ist die zweite Lösung effizienter, da die Benutzung des DMA-Busses vermieden wird.

Beim externen Gerät ist der DMA-Bus durchgeschleift, so daß das System zusammen mit anderer DMA-Peripherie (Festplatten, Laserdrucker) betrieben werden kann. Die eigentliche Vernetzung der verschiedenen Knoten untereinander geschieht über zweiadrige Kabel, wobei Koaxial- oder Twisted-Pair-Verbindungen zum Einsatz kommen. Ebenfalls möglich ist die Verwendung von Lichtwellenleitern. Der Hersteller bietet eine Reihe von Zusatzgeräten zur Signalumsetzung von Draht auf Lichtwellenleiter und zur Signalverstärkung für lange Verbindungen an.

Die physikalische Verkabelung der Knoten untereinander geschieht in Bus-Manier, d.h. die Stationen werden einfach nacheinander geschaltet und erhalten am ersten und letzten Gerät einen Widerstand zum Busabschluß. Durch simples Durchschleifen der Verbindung bleibt die Verkabelung auch durch einen abgeschalteten Knoten bestehen.
x Die Netzhardware sorgt dafür, daß die Kommunikation für industrielle Anwendungen ausreichend störungsfrei arbeitet. Laut Hersteller GTI hat sich das System in der Anwendung in einer Fertigungsstraße bewährt.

Für die Verkopplung verschiedener Rechnertypen stehen entsprechende Netzknoten bereit. So werden Geräte für ATARI ST, PC, VMEbus-Systeme und ECB-Systeme angeboten, allerdings wird z.B. die PC-Anwendung erst circa mit Erscheinen dieses Heftes in Serie gehen. Weiterhin ist es möglich, Rechner per V24- bzw. RS232C-Schnittstelle in das Netz einzubinden.

eLAN und TOS

Für den ST-Anwender wird eLAN praktisch nur beim Booten sichtbar. Das Programm *ELAN.TOS* enthält die komplette Netz-Software und installiert sich im Betriebssystem so, daß das Netz und da-

mit die anderen Stationen als Laufwerk N: angesprochen werden kann. eLAN arbeitet dabei transparent, so daß alle Netzzugriffe ohne Besonderheiten mit den gewohnten Funktionen ablaufen. Auf dem Desktop sieht das z.B. wie in Bild 2 aus.

Beim Booten wird eine Konfigurationsdatei eingelesen, die sämtliche Voreinstellungen enthält. Es handelt sich um eine ASCII-Datei, in der mit Schlüsselworten Installationswerte gesetzt werden. Um beispielsweise den weiter unten beschriebenen Empfang von Nachrichten über das Netz zu unterbinden, müßte hier eine Zeile "MESSAGE=OFF;" stehen. Öffnet man im Desktop Laufwerk N:, erhält man ein übliches Directory, nur daß es sich auf dieser Ebene noch nicht um Dateien, sondern um eine Liste aller angeschlossenen Stationen handelt. Im Beispiel haben diese die Namen *ELANDEMO.xxx*. Öffnet man nun einen solchen "Ordner", befindet man sich im Dateisystem der entsprechenden Station. Im Bild zeigt das untere Fenster nun alle Einheiten an, die von der Station bereitgestellt werden. Hier sind dies die Partitionen C: bis E: und der Drucker (PRN), der an den Rechner *ELANDEMO.002* angeschlossen ist. Welche Einheiten über das Netz bekannt gemacht werden, legt jede Station in der Konfigurationsdatei fest. Es könnte auf *ELANDEMO.002* auch eine Partition G: geben, sie wäre dann aber dem Netz nicht bekannt.

Das dritte Fenster im Bild zeigt die geöffnete Partition D: auf der Station *ELANDEMO.002*. Ab hier hat man es mit tatsächlichen Dateien und Directories zu tun, gerade so, als ob am eigenen Rechner eine weitere Festplatte angeschlossen ist. Das Kopieren einer Datei von *ELANDEMO.002* über das Netz auf den eigenen Rechner ist mit den normalen Desktop-Operationen möglich. Das Starten eines Programms, das auf einer anderen Station abgelegt ist, geschieht ebenfalls wie gewohnt per Doppelklick.

Für ein Anwendungsprogramm bedeutet dieser transparente Zugriff, daß keinerlei Modifikationen notwendig sind. Soll Wordplus einen Text über das Netz von einem anderen Rechner laden, muß lediglich in der gewohnten Dateiauswahl auf das Laufwerk N: geschaltet werden. Innerhalb eines Netzes mit mehreren Benutzern muß es eine Zugriffsregelung geben, da ja nicht unbedingt alle Daten für jeden erreichbar sein sollen. Bei eLAN bestimmt jede Station per Konfigurationsdatei die Zugriffsrechte anderer Stationen auf die eigenen Daten. Mögliche Rechte sind Lesen, Schreiben und der freie Zugriff mit Lesen/Schreiben.

Die Rechte und die dazugehörigen Pfadnamenwerden für die angemeldeten Einheiten vergeben - z.B. Festplatten-Partitionen oder Drucker. Auf diese Weise lassen sich auch einzelne Directories auf den Einheiten freigeben.

Weiterhin ist auch eine Selektion der Stationen möglich, für die Rechte vergeben werden. So könnte man einer Station, die z.B. Backups macht Lesezugriff auf sämtliche Dateien geben, die anderen Benutzer nur an bestimmte Programme und Daten heranlassen.

```
N:\
0 Bytes in 3 Dateien.
  ELANDEMO 002        10-08-89  15:16
  ELANDEMO 003        10-08-89  15:16
  ELANDEMO 006        10-08-89  15:16

N:\ELANDEMO.002\
0 Bytes in 5 Dateien.
  C                   10-08-89  15:16
  D                   10-08-89  15:16
  E                   10-08-89  15:16
  F                   10-08-89  15:16
  PRN                 10-08-89  15:16

N:\ELANDEMO.002\D\
3535 Bytes in 6 Dateien.
  ELANEWP                    24-
  ELANTOS                    24-
  GRAPHIK                    31-
  TEXT                       31-
  ELAN      DAT      3509    26-
  TEST      TST        26    04-

DISKSTATION  DISKSTATION  SYSTEM  DTP  SCRATCH  MWC  eLAN - ST  PAPIERKORB
```

***Bild 2:** eLAN wird als Laufwerk N: angesprochen und gibt Infos über Laufwerke und Knoten genauso wie Dateien aus.*

Da eLAN transparent arbeitet, ergeben Lese- oder Schreibzugriffe, für die keine Berechtigung existiert, normale TOS-Fehler. Anders als gewohnt kann es Directories geben, die zwar in einer Auswahlbox erscheinen, jedoch aufgrund fehlender Berechtigung nicht geöffnet werden können.

Accessory für Servicedienste

Zur Arbeitsunterstützung und für Systemdienste kommt das Accessory *ELAN.ACC* (Bild 3) zum Einsatz. "Fehlerabfrage" zeigt den Netzzustand an; eine reine Service-Funktion. Will man seine Station vor fremden Zugriff bei Abwesenheit schützen, läßt sie sich mit "Sicherung" mit einem Paßwort verriegeln (Bild 4). Es ist dabei möglich, auch Zugriffe über das Netz zu verbieten. Bei der Rückkehr an den Rechner muß man mit dem entsprechenden Paßwort den ST entsichern.

Nach Änderungen an der Konfigurationsdatei der Station muß ein Kalt- oder Warmstart ausgelöst werden. Einige Einstellungen zum Verhalten des Accessories lassen sich auch direkt vornehmen. Wie schon aus dem obigen Beispiel zu ersehen, ist es möglich, den Drucker als "Datei" für das Netz freizugeben. Auf der Station, die den Drucker bereitstellt, kann ein Spooler - UNIX-like bei eLAN "Dämon" genannt - eingerichtet werden, der die ankommenden Daten puffert und nacheinander ausdruckt. Auf diese Weise ist es auch möglich, den ATARI-Laserdrucker per EPSON- oder Diablo-Emulator kostensparend von mehreren STs aus zu benutzen.

Zur direkten Kommunikation zwischen Netzbenutzern können Nachrichten verschickt werden. Dazu stehen im Accessory einfachste Ediermöglichkeiten in einer Dialogbox zur Verfügung. Der Empfänger einer Nachricht ist im einfachsten Fall eine Station. Trifft eine Nachricht ein, wird der dortige Benutzer per Alertbox informiert und kann sich den kurzen Text mit dem Accessory ansehen.

Neben den Stationsnamen existieren, wie zu sehen, auch noch Gruppen. Eine Gruppe ist eine Reihe von Stationen, die logisch zusammengefaßt werden. Das Accessory bietet die Möglichkeit, eine Nachricht auf einen Schlag allen Stationen einer Gruppe zukommen zu lassen.

Die letzte Funktion "Dateitransfer" bietet komfortable Dateifunktionen. Sie könnten zwar auch mit den Desktop ausgeführt werden, das Accessory bietet jedoch mit einfacher Dateiauswahl und leichtem Wechsel zwischen Laufwerken eine nützliche Bequemlichkeit.



Bild 3: Das Accessory ELAN.ACC dient zur Arbeitsunterstützung und für Systemdienste.

eLAN für Programmierer

Neben der reinen Anwendung von "normalen" Programmen und des Service-Accessories im Netz, ist es auch möglich, eLAN in eigene Programme einzubinden. Dazu werden diverse Header-Dateien und Beispielprogramme mitgeliefert. Die C-Quellen wurden mit dem Mark-Williams-System entwickelt.

Auf GEMDOS-Ebene, bei der mit Dateien umgegangen wird, müssen in einem Netzwerk kollidierende Zugriffsversuche vermieden werden. Zwei Programme dürfen nicht gleichzeitig in einer Datei lesen und schreiben, da somit die gelesenen Daten nicht mehr konsistent sind. Damit ist nur erlaubt, daß zwei oder mehrere Programme gleichzeitig in einer Datei lesen. Alle anderen Kombinationen werden von eLAN durch Abweisen eines zweiten Öffnungsversuchs vermieden.

Das Sperren einer kompletten Datei macht aber nicht unbedingt Sinn, wenn zwei Stationen in unterschiedlichen Bereichen schreiben und lesen wollen, die sich jedoch nicht überlappen. Es würde sogar Anwendungen, bei denen mehrere Rechner Zugriff auf eine große Datenbank haben sollen, stark behindern. Daher gibt es das Konzept des Record-Lockings, bei dem ein Programm jeweils nur einen Teil einer Datei für den Zugriff durch andere sperrt. In eLAN gibt es dafür zwei weitere GEMDOS-Funktionen (*Flock* und *Funlock*), die genau diese Funktion übernehmen. Ein netzwerkfähiges Programm müßte entsprechend programmiert sein und z.B. die Datei in dem neuen Zugriffsmodus *SHARED* öffnen.

Die logisch "tieferen" eLAN-Funktionen sind über den XBIOS-Trap unter Benutzung der Funktionsnummern 99-173 eingebunden. *ELAN.TOS* greift also recht tief in das Betriebssystem ein und fügt weitere Funktionen für den Programmierer hinzu.

Die Funktionsgliederung entspricht einer internationalen Spezifikation, dem MAP-Protokoll. MAP steht für "Manufacturing Automation Protocol", das eine echtzeitfähige Architektur für Industrieanwendungen nach dem ISO/OSI-Referenzmodell darstellt. Die verschiedenen Schichten implementieren eine logisch abgestufte Funktionalität und kommunizieren über definierte Schnittstellen.

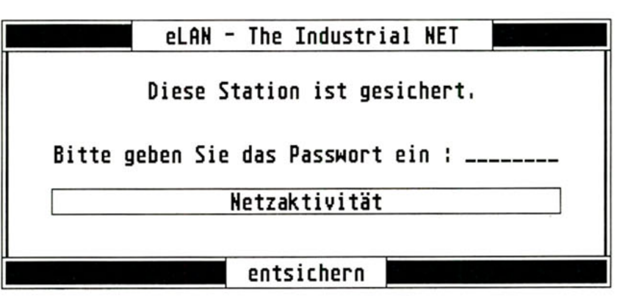


Bild 4: Auch eine Paßwortsicherung ist eingebaut.

Entsprechend kann der Programmierer die Funktionen der einzelnen Schichten benutzen. Ganz unten steht die FIO-Ebene, auf der direkt mit dem Netzknoten kommuniziert wird. FIO steuert den Netz-Controller. Darüber liegt die "Medium Access Control"-Ebene (MAC), die die einfachen Dienste des Netzknotens ansteuert. Es können einfache Datentransfers und interne Netzfunktionen angesteuert werden.

Eine weitere Stufe darüber liegt die "Proway Link Control"-Ebene (PLC), die nicht mehr nur den Austausch von Datenpaketen zwischen Rechnern erlaubt, sondern auch zwischen verschiedenen laufenden Programmen unterscheiden kann. Auf dem ATARI ST können somit z.B. mehrere Accessories und ein laufendes Programm Netzdienste anfordern, ohne daß Daten zwischen den verschiedenen Programmen vertauscht werden.

Schließlich bleibt eine weitere Schicht - dem "Manufactoring Message Format Standard" (MMFS) -, die im ISO/OSI-Modell dem Application-Layer entspricht. Damit sich Programme auf verschiedensten Rechnern austauschen können, müssen Protokolle vereinbart werden, die die bekannten Probleme wie Byte-Sex oder Zeichenkodierung lösen. MMFS stellt Dienste bereit, um zu sendende Daten entsprechend zu konvertieren.

eLAN hält sich mit dieser Aufteilung an die internationalen Spezifikationen und Implementierungs-Richtlinien MAP und PROWAY-C. Für die Programmierung von entsprechenden Anwendungsprogrammen - z.B. einer vernetzten Maschinensteuerung - sind natürlich genaue Kenntnisse der Standards vonnöten, auch im Handbuch wird mehrfach auf die Originalpapiere verwiesen.

Die Dokumentation ist in zwei Ringordner gepackt, und zwar in ein "Technisches Anwenderhandbuch", das die Installation und die Funktionen des Netzcontrollers beschreibt und ein "Software-Anwenderhandbuch", in dem sich die nötigen Informationen zur Software-Konfiguration und den Aufrufen der Netzwerk-Funktionen finden. Die Unterlagen sind detailliert, jedoch ist ein gewisses Maß an Wissen über Netzwerke notwendig. Dies entspricht dem anvisierten professionellen Anwenderkreis.

Token-Passing

eLAN arbeitet nach dem Token-Passing-Prinzip. Dabei wird ein Token auf dem Bus von Station zu Station weitergereicht. Das Token ist eine Art Berechtigungskarte, von der nur eine im Netz existiert und deren Besitzer besondere Rechte gegenüber den anderen Teilnehmern hat. Der Token-Inhaber wird temporär zum Netz-Master und darf damit als einziger senden, wodurch Sende-Kollisionen auf dem Netz ausgeschlossen sind.

Da jede Station sein Token nach einer festgelegten maximalen Zeit weitergeben muß, ist das Verfahren fair und sichert zudem eine feste maximale Zeit, in der eine Station das Token erneut erhält. Damit ist auch eine Voraussetzung für Echtzeit-Anwendungen gegeben.

Da auf einem Bus keine vorgegebene Reihenfolge wie bei einer physikalischen Ring-Vernetzung besteht, wird ein logischer Ring eingerichtet, der die Weitergabe des Tokens bestimmt. Jede Station meldet sich im Netz mit einer Stationsnummer an; die Netzknoten geben das Token in aufsteigender Reihenfolge weiter. Existieren Stationen mit den Nummern 14-1-24-3 und sind diese physikalisch in dieser Reihenfolge verbunden, wird das Token im logischen Ring 1-3-14-24-1-... weitergereicht. Die Stationsnummern können für jede Station frei konfiguriert werden, der Netz-Operator muß allerdings darauf achten, daß keine Nummer doppelt verwendet wird.

Die bei Netzwerken angegebene Übertragungsrate ist üblicherweise ein Bruttowert, der das Übertragen von Protokoll-Informationen und das komplette Netz-Handling mit einschließt. Nach Angaben des Herstellers liegt die effektive Übertragungsrate z.B. beim Dateienkopieren in einem normal ausgebauten eLAN-Netz bei ca. 12 bis 15 kByte pro Sekunde. Vergleicht man das Laden eines Programms über das Netz mit dem gewohnten Verhalten des STs bei direkt angeschlossener Festplatte, fallen keine nennenswerten Geschwindigkeitsverluste ins Auge.

Preise

Die Preisgestaltung für eLAN richtet sich natürlich nach dem Ausbau. Die Hardware kostet pro Netzknoten DM 2262,90 als DMA-Subsystem und DM 1881,- für die Einsteckversion für MEGA-STs. Ein komplettes Kabel über 2 m schlägt mit DM 51,30 zu Buche.

Komplizierter wird's bei der Software. Die Pakete unterscheiden sich in der Anzahl der vorgesehenen Stationen. So kostet die Netzwerk-Software für vier Rechner DM 1687,20, für acht DM 2827,20 und im Vollausbau für bis zu 254 Rechnern DM 5107,20. Die Versionen für das TOS030 des zukünftigen ATARI TT kosten jeweils DM 570,- mehr.

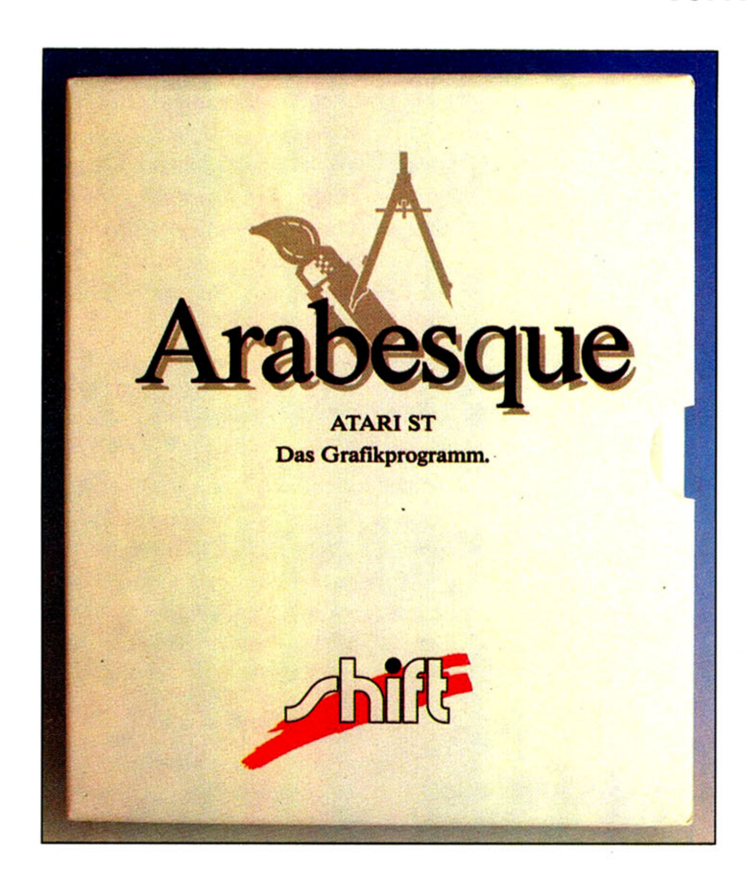
Es ist möglich, eine einmal gekaufte Lizenz auszuweiten, allerdings wird dafür ein Aufschlag von 10% verlangt. Schließlich werden "Starterkits" angeboten, die allerdings keinen Preisvorteil bieten. So kostet die Verbindung mit zwei MEGA-Stationen zusammen DM 5449,20, eine ST-PC Kopplung mit eLAN DM 7216,20.

Der konkrete Preis für eine geplante Netzanwendung kann natürlich nicht pauschal angegeben werden, er ergibt sich aus der unterschiedlichen Anzahl von Netzknoten, aus den verschiedenen Rechnertypen und schließlich aus den individuellen Bedürfnissen an Leitungslänge und Übertragungsmedium.

RT

Bezugsadresse:

GTI
Gesellschaft für technische Informatik mBH
Unter den Eichen 108a
1000 Berlin 45



Arabesque

Ein Grafikprogramm zwischen allen Stühlen?

Verzweifeln Sie nicht, verehrte Leser, dies ist nicht schon wieder ein Test eines Malprogramms. Zugegeben, auch ein Test über ein Malprogramm. Aber Arabesque ist viel mehr. Es ist, als ob mein Stoßseufzer zu Ende des Omikron.Draw!-Testes im letzten Heft Beachtung gefunden hätte - endlich ein Grafikprogramm, das Vektorgrafik beherrscht und damit für Illustrationen von vernünftiger Qualität geeignet ist.

Zur Erklärung: Mit Malprogrammen erzeugte Illustrationen sind immer an die Original-Auflösung gebunden, Ausdrucke in verschiedenen Formaten sind nur mit großen Qualitätseinbußen oder Größenunterschieden möglich. Dies ist im Zeitalter von hochwertigem Textdruck mit Laserdruckern nicht sehr befriedigend. Ein weiterer wichtiger Punkt ist, daß mit Malprogrammen entworfene Schemazeichnungen und Illustrationen oft nur mit unverhältnismäßig viel Aufwand verändert werden können. Da ein Malprogramm eigentlich nicht weiß, was es auf der 'Leinwand' tut, muß wie auf dem Zeichenblatt radiert und korrigiert werden, mit all den unerwünschten Nebeneffekten und Fehlermöglichkeiten, die dieser Tätigkeit innewohnen. Wieviel angenehmer wäre es doch, wenn man einen Kreis als Kreis behandeln und verändern könnte, ohne jedes Pixel einzeln löschen zu müssen, ein Linie nach Belieben dicker oder dünner machen zu können, und all das mit erheblich weniger Mühe als in einem Malprogramm... Nachteil der traditionellen Vektorgrafik-Programme: Für künstlerische Zwecke sind sie im allgemeinen ungeeignet, da sie keine Freihand-Zeichnungen mit Pinseln oder Sprühdosen erlauben. Die bisher auf dem ST verfügbaren Vektorgrafik-Programme hatten den Nachteil, in Handling und Ausgabe limitiert und unhandlich zu sein. GemDraw und EasyDraw waren im wesentlichen an der ersten Generation der Mac-Zeichenprogramme orientiert. Auf dem Mac haben sich Vertreter ihrer Gattung aber inzwischen zu Programmen entwickelt, die selbst Freihand-Zeichnungen mit dem Komfort von Vektorgrafik möglich machen und sogar PostScript-Ausgabe erlauben. Arabesque ist mit all diesen Programmen nicht vergleichbar; es ist jedoch für den ST geradezu revolutionär, da es zum einen einen vollständigen Rastergrafik-Teil enthält, der es mit den meisten üblichen Malprogrammen aufnehmen kann. Zum anderen gibt es einen vollständigen Satz von Vektorgrafik-Funktionen. Damit entwickelte Vektorbilder können auch in Bitmap-Bilder umgewandelt werden. Recht schnell zeigt sich nach dem Start von Arabesque, daß hier keineswegs nach dem Motto 'Nichts halbes und nichts ganzes' verfahren wurde; die beiden Teile von Arabesque können sich beide allein schon gut sehen lassen, zusammen natürlich erst recht.

Rasterbilder

Arabesque bietet so ziemlich alles, was man von einem Malprogramm erwartet: Freihand-Zeichnen, Linien, Linienzüge, Kreis- und Ellipsenbögen, Rechtecke mit scharfen und gerundeten Ecken, Kreise, Dreiecke, Parallelogramme und Polygone. Natürlich kann man auch radieren, mit der Sprühdose hantieren und Flächen füllen. Die üblichen Zeichenparameter wie Linienbreite, Stil, Zeichenmodus, Pinselform, Füllmuster usw. gibt es selbstverständlich auch. Natürlich ist ebenfalls eine Textfunktion eingebaut, so etwas gehört schließlich dazu. Eine Besonderheit beim Zeichnen der Polygone, Dreiecke und Parallelogramme: Wenn die Grundrißfigur fertig ist, kann man optional eine 3D-Figur erzeugen (siehe Bild 6). Das ist für viele Illustrationen eine echte Erleichterung.

In Sachen Rastertext ist Arabesque recht flexibel: Es können GEM-und Signum-Fonts verwendet werden, wobei letztere aber zuerst mit einem Zusatzprogramm in GEM-Fonts konvertiert werden müssen. Die Rasterfonts werden auch im Vektor-Teil verwendet, so daß Schrift in Arabesque nie durch echte Vektorfonts dar-

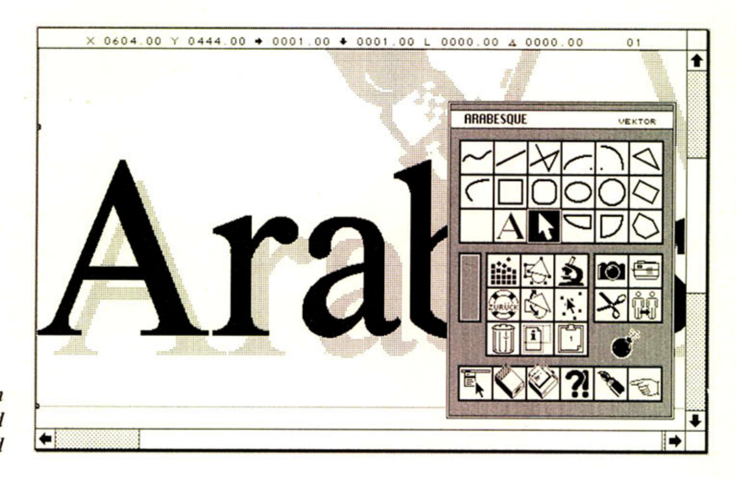


Bild 1: Kombination von Vektor- und Rasterbild

gestellt werden kann. Allerdings erwartet das Programm, daß Rasterfonts für die Verwendung im Vektor-Teil in einer der angewählten Druckerauflösung entsprechenden Form vorliegen (d.h. unterschiedliche Fonts für 9-Nadel und Laserdrucker). Speziell bei Signum ist das natürlich kein Problem. Textblöcke können formatiert werden, wobei sogar Blocksatz zur Verfügung steht. Auch können alle Zeichen des Atari-Zeichensatzes per Alternate-Tastenkombination erreicht werden. Attribute wie Fett- oder Kursivschrift sind vorhanden, die Schreibrichtung kann in 90°-Schritten gedreht werden.

Auch die Füllfunktion bietet einige Spezialitäten: Ein beliebig geformter Bereich kann ganz gewöhnlich mit einem Muster gefüllt werden. Das ist nichts Besonderes. Als Muster läßt sich auch ein beliebig großer vorher ausgeschnittener Block verwenden, man ist also nicht auf das kleine Musterformat von GEM beschränkt. Sehr praktisch und wenig verbreitet ist dagegen die Verlaufsfüllfunktion, die eine beliebige Fläche mit einem Graustufenverlauf auffüllt. Besonders hübsch ist schließlich die 'Block einsetzen'-Funktion. In diesem Modus wird ein vorher ausgeschnittener Block soweit verformt, daß er genau einmal in den zu füllenden Bereich hineinpaßt, dies bildet einen Gegensatz zum oben erwähnten Füllmodus, in dem der Block ohne Veränderung einfach als Muster verwendet wird. Schnell und komfortabel lassen sich so die wildesten Blockverformungen erzeugen. Natürlich gibt es hierfür auch durchaus sinnvolle Anwendungen: Z.B. ein Etikett auf eine frisch gezeichnete Flasche zu 'kleben', wird erheblich vereinfacht. Natürlich kann man, wie in jedem Malprogramm, das etwas auf sich hält, auch in Arabesque-Blöcke auf sehr viel vielseitigere Art verändern. Spiegeln, Vergrößern, Drehen, Invertieren, mit beliebigen Kurven biegen, perspektivisch verzerren, Scheren, mit Schatten versehen usw. Besondere Beachtung verdienen noch die Funktionen zur Herausarbeitung von Konturen, zur 'Einfärbung' mit einem Muster und zum Glätten von allzu stufigen Vergrößerungen. Kopieren und Verschieben wurde natürlich nicht vergessen, eine UNDO-Funktion hilft bei der Rücknahme unerwünschter Operationen. Die Blockfunktionen glänzen auch durch Schnelligkeit. Eine Lupe und zusätzliche Seiten, soviele der Speicher hergibt, sind ebenfalls vorhanden. Sehr positiv sind die freie Auswahl des Bildschirmformates und die Fähigkeit, auch DIN A3-Formate zu bearbeiten (falls ein geeigneter Drukker zur Verfügung steht). Eine Ganzseitenübersicht hilft, das Layout zu beurteilen. Sehr nützlich ist eine Funktion zur Begrenzung der Zeichenarbeit auf bestimmte Ausschnitte, um ungewollte Effekte zu vermeiden.

Außenwelt

Arabesque kann - auch das gehört sich so - die gebräuchlichen Bildformate wie Degas, Stad, Image und IFF laden. Natürlich gibt es auch ein eigenes Format. Praktischerweise kann Arabesque Fremdformate nicht nur lesen, sondern auch schreiben. Auch Pinsel, Muster und sonstige selbstdefinierte Arbeit können gesichert werden. Wenn wir nun schon bei der Außenwelt sind: Die von Arabesque les- und schreibbaren Vektorgrafikformate sind ein eigenes und das GEM-Metafile-Format, das auch von Calamus verstanden wird. Natürlich gehört auch die Welt des Druckes hierher. Arabesque verwendet normalerweise Druckertreiber, die, wie die Hex-Dateien von WordPlus, per Editor veränderbar sind. Für die wichtigsten Drucker werden Anpassungen mitgeliefert. Einige Drucker, die sich auf diesem Wege nicht vernünftig ansteuern lassen, so zum Beispiel der ATARI-Laserdrukker, werden gesondert behandelt.

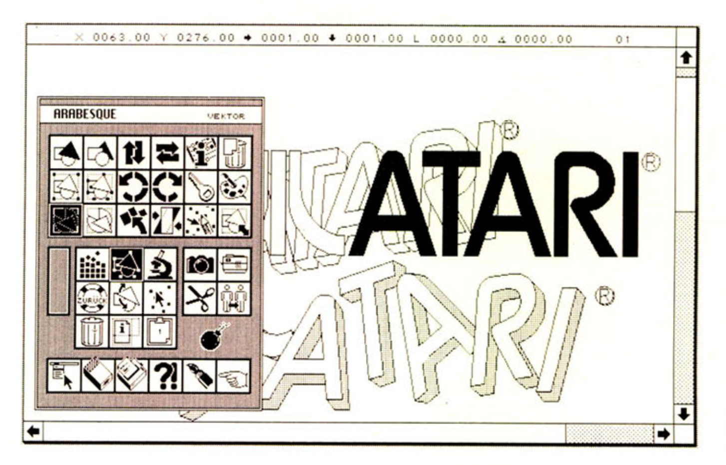


Bild 2: Blockfunktionen im Vektorteil

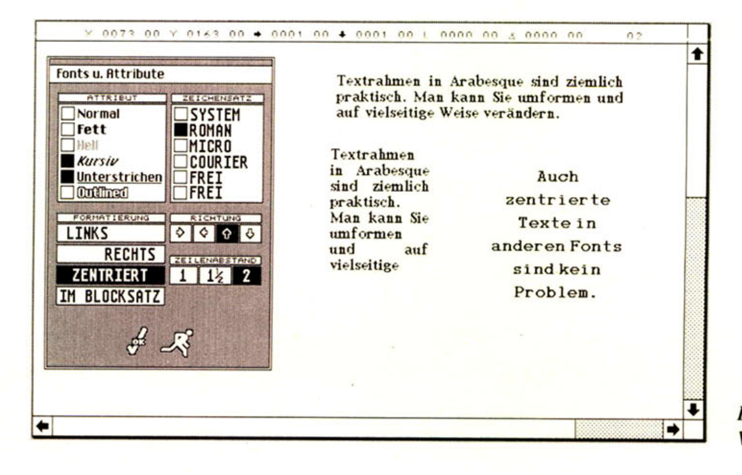


Bild 3: Vektortextfunktion

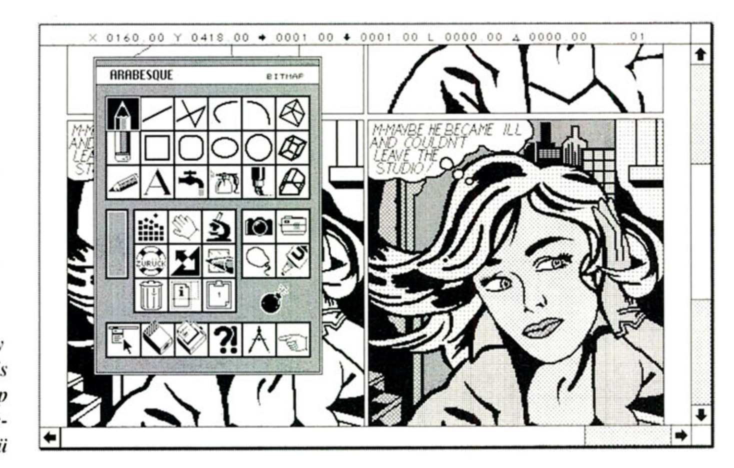


Bild 4: Roy Lichtenstein als Arabesque-Bitmap und das Arabesque-Bitmap-Menü

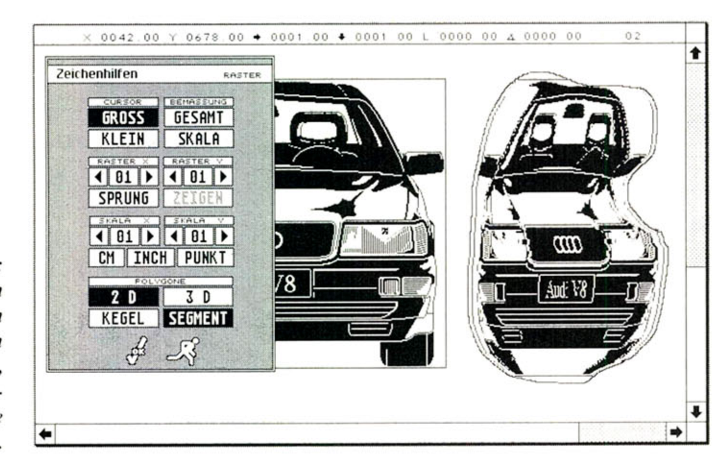


Bild 5: Hilfsfunktionen zum Rasterzeichnen. Im Hintergrund ein viereckiger Block, der in eine Freihand-umrandete Fläche eingesetzt wurde.

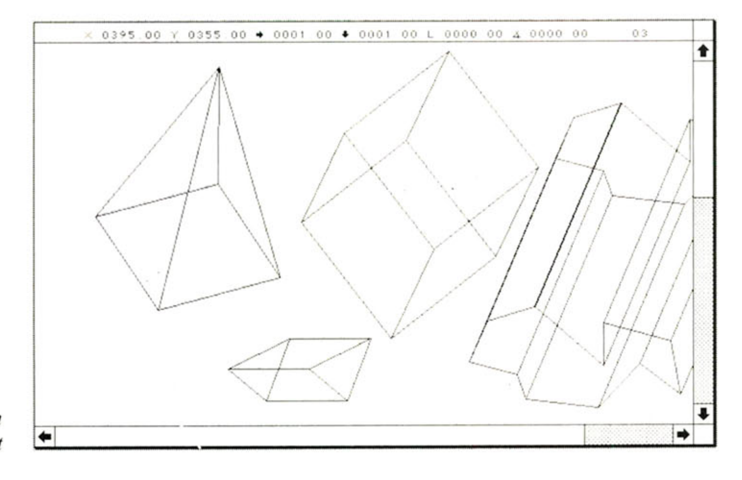


Bild 6: 3D-Formen einfach erzeugt

Vektorgrafik

Die Zeichenfunktionen im Vektor-Teil von Arabesque unterscheiden sich im Umfang nicht allzusehr von ihren Kollegen im Rasterbereich. Natürlich gibt es keine Sprühdose und kein allgemeines Flächenfüllen, denn diese Funktionen sind in Vektorgrafik nur schwer zu implementieren. Leider wurden die Funktionen für Dreiecke, Parallelogramme und Polygone nicht mit der 3D-Option des Rasterteils ausgestattet. Auch die Möglichkeiten zum Freihand-Zeichnen sind stark limitiert: Im Freihand-Modus kann man per Maus Linienzüge von bis zu 512 Punkten erzeugen. Bei Vergrößerungen erkennt man sofort, daß es sich nicht um Kurven, sondern um Linienzüge handelt, die eben aus geraden Stücken zusammengesetzt sind. Was Arabesque in diesem Zusammenhang eindeutig fehlt, sind interpolierende Kurven in irgendeiner Form, ob Bezier-Kurven (wohl die verbreitetste Form), Akima-Interpolationen oder Beta-Splines. Solche Kurven werden über ein paar Kontrollpunkte kontrolliert und können beliebig vergrößert werden, ohne ihre Kurveneigenschaften zu verlieren.

Sehr praktisch ist die Vektor-Textfunktion: Zwar werden hier, wie bereits oben erwähnt, keine Vektorfonts verwendet, der Text wird aber dafür in einen Rahmen geschrieben, in dem er, ähnlich wie in Calamus, formatiert wird, und zwar rechts-, linksbündig, zentriert oder im Blocksatz. Gerade bei der Textbearbeitung zeigen sich die Vektoreigenschaften von Arabesque sehr deutlich: Der Textrahmen kann jederzeit wieder angewählt, der Inhalt verändert oder umformatiert werden. Auch die Rahmengröße und Position ist zu jedem Zeitpunkt veränderbar. Natürlich gilt das auch für jedes andere Objekt: Verformen, Ändern von Attributen etc. ist kein Problem, weil das Programm jedes gezeichnete Objekt 'kennt' und nicht nur die Repräsentierung durch Punkte auf dem Bildschirm bearbeitet. Leider gibt es keine Möglichkeit, die Größe von Objekten direkt per Zahlenkoordinaten zu bearbeiten. Auch fehlt ein Lineal für exaktes Arbeiten. Aber trotzdem ist der Vektorteil für Illustrationen viel geeigneter als jedes Malprogram. Die Zeit, die man sonst angestrengt mit der Lupe hantiert, um irgendwelche irrtümlich gezeichneten Pixel zu löschen, kann man wahrlich besser nutzen. Die meisten Grafikobjekte lassen sich auch noch weitergehender ändern: Man kann zum Beispiel Punkte zu Polygonen hinzufügen oder aus denselben löschen. Ein Gruppenmechanismus erlaubt es, mehrere Objekte gemeinsam zu bearbeiten, was den Komfort erheblich erhöht.

Wichtig bei Vektorgrafik ist auch, daß der Benutzer bestimmen kann, auf welche Art sich Objekte überdecken. Stellen Sie sich die gezeichneten Objekte als ausgeschnittene Papierstücke vor, die sich auf dem Schreibtisch überlappen. Sie können in Arabesque jedes Objekt in den Vorder- oder Hintergrund verschieben, bis Sie die gewünschte Überlappung erreicht haben. Weitere Möglichkeiten sind Spiegeln, Rotieren, Scheren und Skalieren. Es gibt hier jedoch einige Beschränkungen, die daher rühren, daß zum Zeichnen TOS-Funktionen verwendet werden. Das TOS kann aber zum Beispiel keine gedrehten Rechtecke mit abgerundeten Ecken zeichnen. Daher werden solche Objekte beim Drehen in normale Rechtecke verwandelt. Sehr nützlich ist es, daß man einzelne Objekte vor Veränderungen schützen kann. Auch im Vektor-Modus gibt es eine Übersichts-Darstellung, allerdings

nur von ungefähr $^2/_3$ der Seite. Wie bereits erwähnt, ist es schließlich auch möglich, die gesamte Grafik in eine Rastergrafik einer bestimmten Größe zu verwandeln. Auch können Rastergrafikblökke in Vektorbildern verwendet werden, mit all den Nachteilen bezüglich Vergrößerungen usw., die Rasterbilder nun einmal mit sich bringen.

Bedienung

Arabesque stellt fast den gesamten Bildschirm zum Zeichnen zur Verfügung. Oben gibt es eine Kopfzeile, die allerdings nicht für Menüs, sondern für die Anzeige von Koordinaten verwendet wird. Rechts und Unten gibt es Schiebebalken. Accessories können aufgerufen werden. Die eigentliche Bedienungsoberfläche des Programms sind zwei große Pop-Up-Boxen, die alle wichtigen Funktionen von jeweils einem Programmteil enthalten. Sie werden von einer Menge Dialogboxen unterstützt. In den Pop-Up-Boxen gibt es oben die Icons für die Zeichen-Grund-Funktionen, in der Mitte all die Bilder für wichtige Arbeitshilfen, und unten schließlich allgemeine Funktionen wie Diskettenoperationen, Drucken usw. Zwischen Menü und Zeichenfläche wechselt man mit der rechten Maustaste. Die Pop-Up-Box verschwindet dann oder erscheint wieder, und zwar dort, wo sich der Mauszeiger gerade befindet. Im Großen und Ganzen ist Arabesque recht leicht zu bedienen, an einigen Stellen sind die Icons allerdings nicht ganz so einleuchtend geraten. Die Maustasten sind meistens konsistent belegt. Sehr ausführlich wird die Tastatur unterstützt. Weniger tierliebe Benutzer werden das begrüßen. Für das Handbuch gilt das gleiche wie für die Oberfläche: Im großen und ganzen gelungen, nicht immer übersichtlich, aber recht hilfreich.

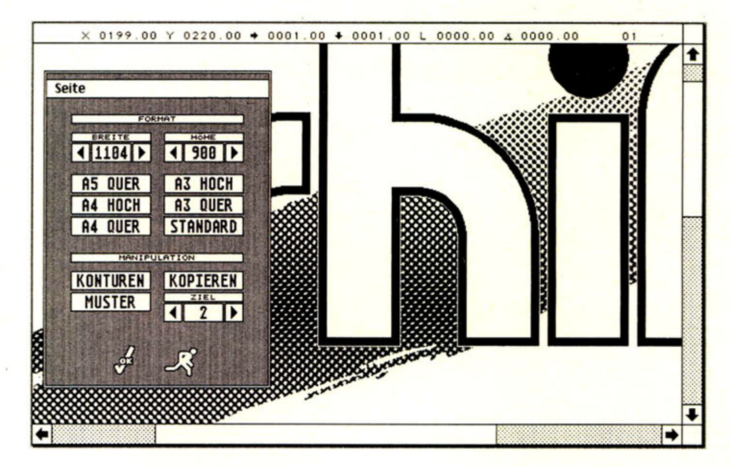


Bild 7: Arabesque-Seitenformat

Zum Schluß

Arabesque ist ein gelungenes Allroundprogramm. Wie gesagt, für Illustrationen zu Artikeln oder Aufsätzen halte ich es für erheblich geeigneter als jedes reine Malprogramm. Einige Möglichkeiten könnte der Vektormodus noch verkraften, besonders Freiform-Kurven fehlen sehr. Die Bedienungsoberfläche ist gelungen und auch übersichtlich, wenn man sich einmal die Bedeutung der Icons gemerkt hat (eigentlich komisch, früher mußte man sich - unwilligerweise - Tastenkürzel merken, jetzt sind es Icons). Für Vielschreiber und als preisgünstiges Illustrationsprogramm (DM 278,-), zum Beispiel für Calamus, ist Arabesque gut geeignet.

CS

Bezugsadresse:

Shift
Sonnenschein und Hansen
Unterer Lautrupweg 8
2390 Flensburg
Tel.: 0461/22828

Aufstieg in die 3. Dimension



DynaCADD erlaubt CAD im Raum

Als Ergänzung zu unserem Bericht über CAD-Programme im Juni-Heft geht es heute um DynaCADD, ein Produkt aus Kanada. Waren unsere Kandidaten im Sommer ausschließlich auf Zeichnungen in der Ebene beschränkt, so erlaubt DynaCADD 'echte' 3D Konstruktionen.

Warum aber die Betonung auf echte? Gibt es etwa eine falsche dritte Dimension? Ja und Nein: Man unterscheidet bei CAD-Programmen solche, die 2D, 22D und 3D beherrschen. Während erstere ausschließliches Zeichnen auf einer Fläche erlauben, sind die 22D-Programme in der Lage, räumliche Modelle in Form von Prismen zu erstellen. Das bedeutet, daß man ein Bauteil zeichnen und ihm eine Tiefe zuordnen kann. Vorder- und Hinterseite sehen also stets gleich aus.

3D-Software schließlich ermöglicht das Zeichnen aller Elemente auf allen Ebenen im Raum. Hier wiederum gibt es Programme, die auch Flächen (sowohl eben als auch gekrümmt) verwalten. Dergleichen ist natürlich mit hohem Rechenaufwand verbunden. Soll z.B. ein auf diese Weise gezeichnetes Gebäude realistisch dargestellt werden, muß das Sich-Überdecken der Elemente betrachtet, das Licht schattiert werden etc... Dieses Verfahren bezeichnet man als *Solid Modeling*. Programmpakete, die so etwas beherrschen, kosten viel Geld und sind rein professionellen Anwendungen im Workstation-Bereich vorbehalten. Selbst die für PCs erhältlichen Programme wie AutoCAD 10.0 sind eher hübsch anzusehen, als damit sinnvoll gearbeitet werden könnte, denn selbst der schnellste 386er AT ist dafür zu lahm.

Von der Fläche zum Draht

DynaCADD erlaubt - und das ist die einfachste Form der 3D-Darstellung - die Erstellung von Drahtgittermodellen. Wie die Abbildungen zeigen, muß man sich die gezeichneten Körper repräsentiert durch ihre Kanten vorstellen: Sie scheinen aus Draht gebaut. Es braucht bei der Darstellung keinerlei Unterscheidung stattzufinden, ob ein Element der Zeichnung vor einem anderen liegt und es eventuell verdeckt. Ebenfalls ist ausschließlich die Parallelprojektion möglich, um einen räumlichen Effekt zu erlangen; Fluchtpunktperspektiven sind ausgeschlossen.

Durch diese Einschränkungen hält sich die Rechenzeit, die DynaCADD für dreidimensionale Zeichnungen braucht, in Grenzen. Trotzdem - das sei vorausgeschickt - nimmt bei komplexeren Gebilden vor allem der Bildschirmaufbau deutlich Zeit in Anspruch. Bevor ich näher auf die Arbeitsweise des Programms eingehe, sollte nicht unerwähnt bleiben, daß mir die Version 1.36 von DynaCADD zum Test zur Verfügung stand. Diese ist allerdings nicht mehr aktuell; während diese ST-Computer gedruckt wird, wird die Version 1.64 ausgeliefert werden. Aber das ist das Schicksal der Presse: Ehe etwas gedruckt wird, ist es schon überholt.

Packen wir DynaCADD zunächst aus: Auf drei Disketten befinden sich Programm, Beispiele und einige kByte Hilfe-Dateien, die die Online-Hilfe ins laufende Programm einblendet. Dergleichen ist zwar im PC-Bereich selbstverständlich, wir ST-Freaks freuen uns aber noch darüber, daß der Griff zur 'Help'-Taste den Blick ins Handbuch ersetzt. In der neuesten Version werden diese Dateien - wie das gut verständliche Handbuch (sehr gutes Stichwortregister!) selbst - auch in deutscher Zunge verfaßt sein. Den Fonteditor, den das Handbuch verspricht, suchte ich leider vergeblich.

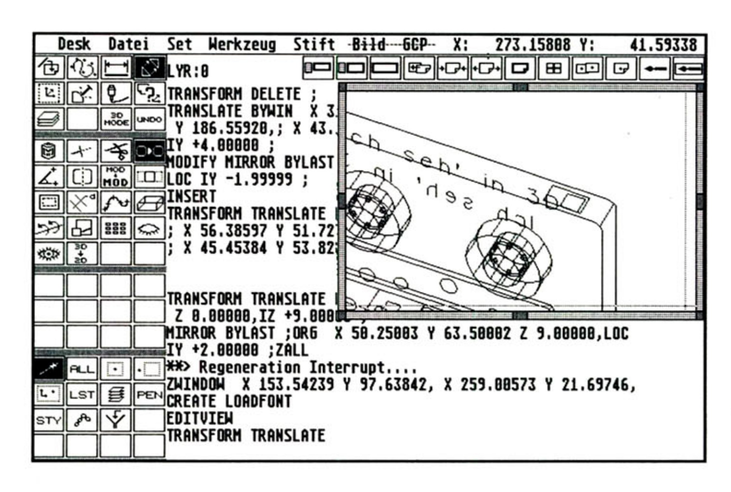


Bild 1: Arbeits- und Textfenster von DynaCADD

Doch da rutscht noch ein kleiner Kunststoffkasten aus der Verpackung: Ein Dongle = Hardwarekopierschutz. Dagegen ist eigentlich nichts einzuwenden, allerdings muß ich gegen das blockieren des Romports aufs energischste protestieren. Hätte nicht auch ein Dongle für den Joystickstecker gereicht? Ich habe nämlich auch noch einen Scanner, einen Aladin, einen Easytizer und eine Profibank und die Buchse links in meinem ST ist schon reichlich ausgeleiert...

Mit über 500 kB Länge ist DYNACADD.PRG ein kräftiger Riemen. Wer große Zeichnungen verarbeiten will, sollte über mindestens 2 MB RAM in seinem ST verfügen, eine Festplatte ist ebenfalls dringend anzuraten. Es geht auch ohne, aber wer richtig mit dem Programm arbeitet (und das werden beim Preis von 2248,- DM sicher alle Käufer tun), der sollte bei der Hardware ohnehin nicht zaudern. Eine ST-Workstation mit Großbildschirm, Harddisk und Plotter schlägt mit ca. 15000,- DM zu Buche; ein Preis der sich bei entsprechendem Einsatz schnell rentiert.

Per Doppelklick gestartet, empfängt einen DynaCADD nach längerer Ladefrist mit der fünfgeteilten Arbeitsfläche. Oben das Menü, darunter die Symbole, die den Blick auf die Zeichnung verändern (Scroll, Zoom). Links die Befehlsauswahl, wobei die verschiedenen Hierarchien eines Kommandos übereinander angeordnet werden. Die Arbeitsfläche selbst besteht aus einem Textfenster, in dem die Arbeit in Klartext protokolliert wird. Darüber liegt - frei verschiebbar - das Zeichnungsfenster (siehe Bild 1). Hier hinein laden wir erst einmal die kleine Zeichnung aus dem 2D-Tutorial.

Freundlicherweise gibt es nämlich hinten im Handbuch eine kleine Lehrveranstaltung: das Tutorial. Für meinen Geschmack könnte es ausführlicher sein. Besonders stellt sich beim Befolgen desselben immer die Frage: 'Warum muß ich diese Schritte durchführen?' Das Konzept des Programms (es hat nämlich eines und sogar ein gutes!) müßte eingangs erläutert sein, dann würde so manches Mißverständnis vermieden.

Dazu ein Beispiel: Wenn DynaCADD die Eingabe eines Punktes (z.B. den Startpunkt einer Linie) erwartet, kann dieser auf so vielfältige Art und Weise eingegeben werden, daß es eine Freude ist. Selbstverständlich ist das Fangen von End-, Mittel-, Schnitt- und Konturpunkten anderer Zeichnungselemente. Aber auch Koordinateneingaben sind - kartesisch oder polar - absolut oder auch relativ zum vorhergehenden Punkt möglich. Dabei erscheint, wie bei allen numerischen Eingaben, ein Taschenrechner (siehe Bild 2). Dieser hat die wichtigsten Funktionen parat und ist eine ungeheure Hilfe beim Zeichnen. Das Konzept dieser doch recht komplexen Punkteingabe wird auch im Handbuch nur angedeutet, das Tutorial leitet den Anwender ohne nach Sinn und Zweck zu fragen, durch die Befehle. Für den Benutzer mit Erfahrung in einschlägiger CAD-Software ist das Konzept der Bedienung sicher kein Problem, doch wendet sich DynaCADD wohl eher an Anfänger in diesem Metier. Es ist z.B. möglich, als Spiegelachse eine Gerade zu benutzen, die - sagen wir - senkrecht auf der Zeichenebene steht und durch einen Kreismittelpunkt geht. Dabei braucht diese Gerade aber nicht gezeichnet zu sein! Die Koordinateneingabe machts möglich.

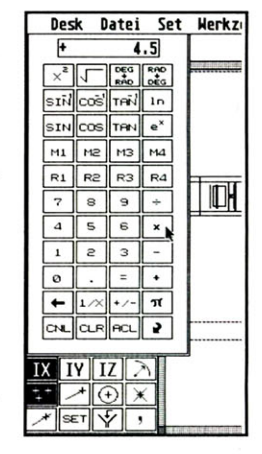


Bild 2: Der Taschenrechner steht bei jeder numerischen Eingabe zur Verfügung.

DynaCADD trennt zwischen 3D- und 2D-Zeichnungen. Bei der Bedienung im 2D-Modus wird ganz 'normal' gezeichnet. Dieser unterscheidet sich im Befehlsumfang kaum vom 3D-Modus, nur liegt die Zeichnung eben in nur einer Ebene. In dieser Gangart ist DynaCADD durchaus als Konkurrenz zu den vier in der Juni-

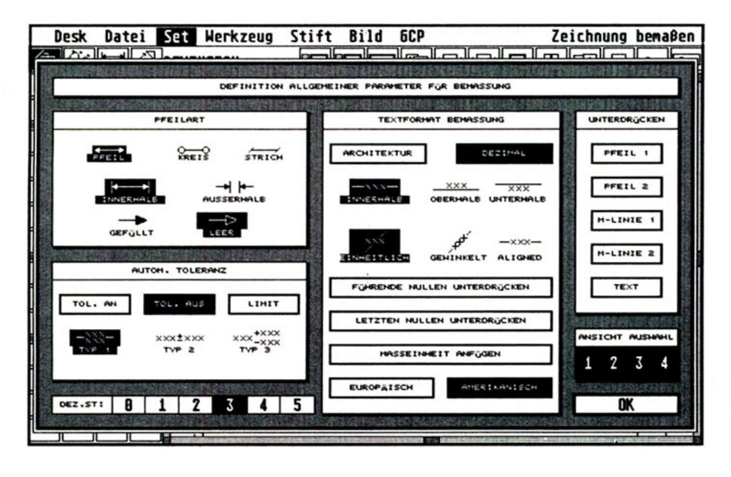


Bild 3: Parameter der Bemaßungsfunktion

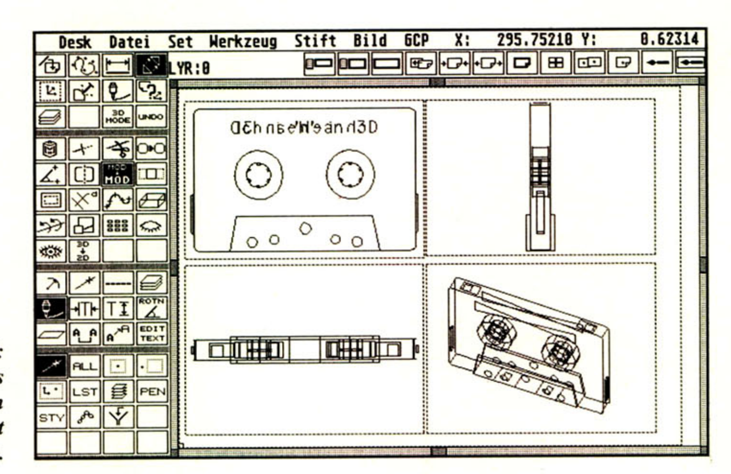


Bild 4: Vier Ansichten eines 3D-Körpers können gleichzeitig bearbeitet werden.

Ausgabe getesten Programmen (TechnoCAD, CADproject, CADja und BeckerCAD) zu sehen. Wenn in der neuesten Version auch Schraffuren enthalten sind, dann ist es vom Befehlsumfang her sogar Spitzenreiter.

Vielfältig sind die Funktionen zum Selektieren einzelner Objekte, zum Zeichnen, zum Verändern... Die Aufzählung wäre endlos, wollte sie vollständig sein. Insgesamt sollen es über 300 Icons sein, die am linken Bildschirmrand erscheinen. Dabei hangelt man sich in bis zu vier Schritten abwärts: Zum Zeichnen einer Linie wählt man zunächst 'Einfügen', dann aus dem zweiten Menü 'Linie', dann 'Horizontale', woraufhin die erwähnte Auswahl zur Angabe des Start- und Endpunktes erscheint. Alle Befehle werden mit der Maus eingegeben, die Formulierung als regelrechten Befehl und dessen Eingabe über die Tastatur ist aber genauso möglich. Mit der Zeit bekommt man darin auch Übung, denn bei der Mauseingabe wird stets der entsprechende Befehl in Textform im Arbeitsfenster gezeigt. Wie gesagt ist die Befehlsvielfalt enorm. Allein die unterschiedlichen Bemaßungsfunktionen sowie deren Parametrisierungsmöglichkeiten (siehe Bild 3) sind eine Pracht und zeugen von Fleiß auf Programmiererseite.

Besonders ragen auch die Spline- bzw. Bezierfunktionen aus dem Standard der ST-CAD-Software heraus. Nachdem diese Kurven gezeichnet wurden, ist eine Änderung durch Verschieben der Stützpunkte immer noch möglich. Nicht so gut gefällt hingegen das Prinzip von DynaCADD, immer erst das fertige Ergebnis am Monitor zu zeigen. Ist man von anderen Programmen her gewohnt, daß beim Verschieben von Objekten diese 'an der Maus hängen', so wird man von DynaCADD enttäuscht. Hier werden ganz klassisch Start- und Zielpunkt plaziert und dann regeneriert das Programm die Zeichnung.

Die dritte Dimension

Kommen wir nun zum Thema des Tages: Dem 3D-Modus von DynaCADD. Wenn man genau überlegt, müßte man sich eigentlich fragen: 'Wie geht das überhaupt? Wie kann ich in gewohnter ST-Manier, nämlich mit der Maus, eine räumliche Zeichnung erstellen, wenn der Monitor doch flach ist?'

Die Frage ist durchaus berechtigt. Es ist auf den ersten Blick tatsächlich schlecht möglich. Der Raum (Vorsicht, es wird jetzt leicht mathematisch!) wird i.allg. durch ein karthesiches Koordinatensystem mit X-, Y- und Z-Achse beschrieben. Diese drei Achsen stehen jeweils senkrecht aufeinander. Wenn dieser Raum auf eine Ebene wie den Monitor projiziert wird, geschieht das am günstigsten so, daß eine Achse senkrecht zu ihm steht. Schauen Sie sich die Bild 4 an, in der die Beispiel-Cassette in vier Ansichten zu sehen ist. Oben links ist die Draufsicht. Hier liegt die X/Y-Ebene parallel zum Monitor, die Z-Achse ist unsichtbar. Wird beispielweise in dieser Ansicht gezeichnet, so fehlt die Information, welche Z-Koordinate die eingegebenen Punkte haben.

Ein Kreis wird eindeutig beschrieben, wenn man die Endpunkte seines Durchmessers bestimmt. Das gilt jedoch nur für die Ebene. Im Raum fehlt noch ein dritter Punkt, der irgendwo auf dem Umfang liegt. Erst durch diesen ist einer der unendlich vielen Kreise eindeutig beschrieben, die durch die beiden ersten Punkte gehen. Konsequenterweise dürfte also das Icon für 'Kreis mittels Durchmesser definieren' im 3D-Modus nicht anwählbar sein. Es sei denn, man legt (ziemlich willkürlich) alle Zeichenbefehle in die X/Y-Ebene, wie es DynaCADD auch tut.

Auf diese Weise zieht sich das Programm also aus dem Schlammassel der fehlenden dritten Dimension. Doch was tun, wenn ich auf Ebenen arbeiten will, die nicht senkrecht zum vorgefertigten Koordinatensystem stehen? Nun, als Ausweg kann sich der Benutzer Koordinatensysteme selbst definieren. Die wichtigsten sind schon fest eingebaut, weitere neun sind frei wählbar. Die Systeme werden wie eine ganz normale Ebene definiert. Die Angabe von drei Punkten oder eines Normalenvektors ist möglich. Ein solches Koordinatensystem muß nur noch aktiviert werden und schon beziehen sich alle Koordinateneingaben auf dieses System. Konsequenterweise bedeutet also im 3D-Modus 'horizontal' nicht 'horizontal auf dem Monitor', sondern lediglich 'parallel zur X-Achse', und zwar zur jeweils gültigen.

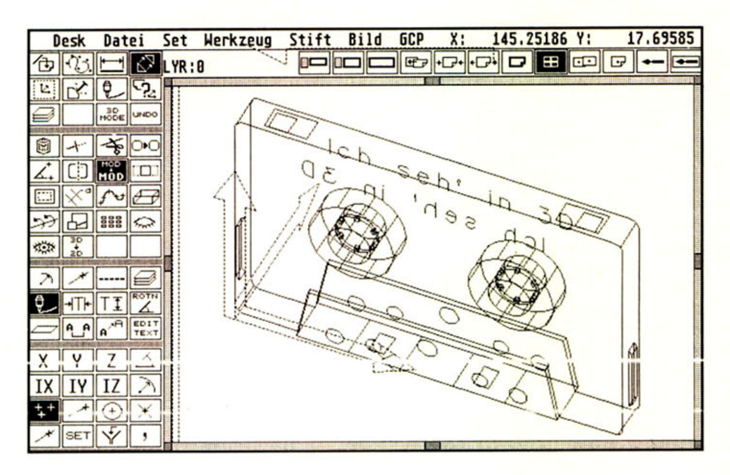


Bild 5: 3D-Modell mit Koordinatensystem

Mit Hilfe der 'Geometric Coordinate Planes', so heißen diese Koordinatensysteme, ist es demnach möglich, mit Dyna-

CADD im Raum so zu arbeiten, wie man es aus dem 2D-Teil gewohnt ist. Doch hier sei das Handbuch zitiert: 'GCPs sind ein mächtiges Werkzeug, doch erfordern sie Geduld und Zurückhaltung'. Das kann nur betont werden. Ist CAD-Software im allgemeinen recht konzentrationsbedürftig, so tritt diese Forderung beim räumlichen Konstruieren erst recht auf. Besonders das permanente Sich-Räumlich-Denken der Zeichnungsteile fällt vielen Menschen schwer. Da die Modelle reine Drahtgittermodelle sind, ist auch keinerlei Anhaltspunkt gegeben, wo vorn und wo hinten bei einer Ansicht ist. Trotz alledem ist der Umgang mit dreidimensionalen Zeichnungen leicht zu erlernen, und das Prinzip der GCPs ist dabei eine große Hilfe.

Arbeit im Raum

Bild 5 zeigt das 3D-Modell 'Cassette' mit einem Koordinatensystem, das sinnvollerweise parallel zu den Körperkanten plaziert wurde. Die Ansicht ist so gewählt, daß keine Achse senkrecht zur Projektionsebene, also dem Monitor, steht. In dieser Ansicht ist per Mausklick natürlich extrem schlecht zu zeichnen. Aber trotzdem erlaubt DynaCADD sinnvolles Arbeiten damit. Da die meisten Elemente der Zeichnung Bezug zueinander haben, kann durch Einrasten auf End- oder Mittelpunkten gut gearbeitet werden.

Trotzdem kann folgende Situation eintreten: Man will die Verbindungslinie zwischen den Mittelpunkten zweier Kreise zeichnen. Der erste ließ sich noch problemlos nach Anwahl des 'Fange Mittelpunkt'-Icons selektieren, doch jetzt geht nichts mehr, weil man den zweiten Kreis in dieser Ansicht nicht eindeutig selektieren kann. Hier hilft nur, den Kreis in einer anderen Ansicht zu selektieren. Also: Zoom auf die gesamte Zeichnung (wie in Bild 4), darin den Zoom für den richtigen Ausschnitt gewählt und jetzt den zweiten Kreis selektiert. Das alles geht, ohne die Funktion 'Linie' zu verlassen! Der Blick auf die Zeichnung kann - auch im 2D-Teil - jederzeit verändert werden, ohne die laufende Funktion zu beeinflussen. Dieses Feature ist bisher unter den CAD-Programmen auf dem ST einzigartig und verdient ein dickes Lob.

Ansonsten verbergen sich hinter vielen Befehlen wahre Kostbarkeiten wie z.B. das 'Revolving'. Mit diesem Befehl werden Zeichnungsteile um einen Vektor im Raum rotierend vervielfältigt. Selbstverständlich sind Anzahl der Kopien und Winkel der Rotation einzustellen (es gibt bei DynaCADD nur sehr wenige Befehle, an denen nichts parametrisiert werden kann). Mittels Revolving waren die Zähne der Bandrollen in der Cassette kein Problem. Entsprechend gibt es das 'Sweeping'. Dabei werden die selektierten Objekte nicht an einem Vektor, sondern entlang eines Pfades (das kann auch ein Spline sein) kopiert. So sind gekrümmte Gitter schnell konstruiert. Diese Befehle sind - wie bereits erwähnt - nur ein Ausschnitt aus dem gesamten Repertoire von DynaCADD. Sie zeugen, wie auch das Erscheinungsbild der Arbeitsfläche usf., von Liebe zum Detail und Perfektionismus.

erfüllten ihre Aufgabe im 3D-Modus nur teilweise und warfen mit unsinnigen Fehlermeldungen um sich. Im Handbuch wurden sie erst gar nicht erwähnt. Absolut unsinnig ist die Logik des Programms, beim Trimmen immer den kürzeren Teil der Linie zu löschen. Andere CAD-Software löscht dabei die zuerst angeklickte Seite.

Viel garstiger jedoch erwies sich die Unwilligkeit von DynaCADD, die Arbeit überhaupt anzutreten. Das Laden der Zeichnungen wurde häufig mit drei Bomben quittiert. Nach einem Reset und erneutem Start ging's dann meist. Weis der Programmierer allein, woran das liegt.

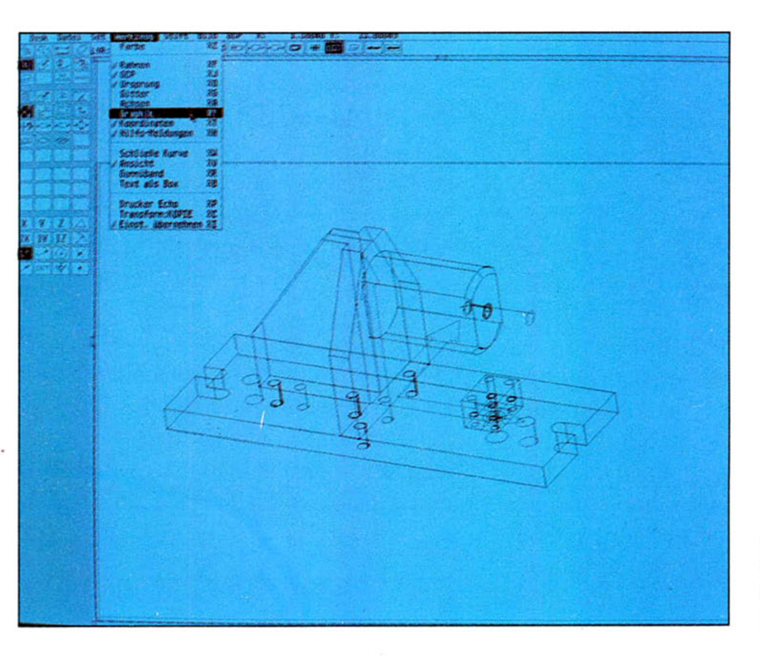


Bild 6: DynaCADD läuft auch mit einem Großbildschirm.

Tricks und Bugs

Es sind all die Kleinigkeiten, die das Arbeiten vereinfachen: Beim mehrfachen Druck auf das Icon 'letzter Ausschnitt' werden nacheinander alle Zoomfenster wieder gezeigt, die man seit Sitzungsbeginn gewählt hatte (und nicht nur das letzte). Per 'Undo' lassen sich zumindest die Änderungen seit dem letzten Hauptbefehl zurücknehmen. Doch wenn die zuletzt gezeichneten Elemente wieder gelöscht oder anderweitig bearbeitet werden sollen, dann gibt's eine edle Hilfe: Das Selektions-Menü erlaubt per Mausklick nacheinander die Anwahl der Zeichnungsteile in umgekehrter Erstellungsfolge.

Trotz allem bleiben auch bei DynaCADD Wehmutstropfen: Die Trimmfunktionen

Eine Plotterausgabe der Cassette hätte ich Ihnen zu gern präsentiert, jedoch erwies sich die Dialogbox, die nach Anwahl des 'Plotter'-Icons erschien, als Sackgasse im Computersinne. Sie hatte nämlich keinen Ausgang und so wurde auch nicht geplottet, sondern der Griff zum Reset-Taster war mal wieder unumgänglich.

Eine Druckausgabe auf NEC-kompatiblen 24-Nadeldruckern, die 360 dpi-Auflösung auch in der Vertikalen besitzen, ist ebenso wenig möglich, wie das Handbuch den Aufbau der Druckertreiber erklärt. Es ist im PC-Bereich wohl üblich, ungeheure Mengen von Drucker- und Plottertreibern mitzuliefern (Wodurch unterscheiden die sich eigentlich außer im Namen?), doch ein paar Worte über den Aufbau hätten niemandem geschadet.

Überhaupt ist zwar auch das Handbuch mit Detailfreude gestaltet, doch für ein Softwarepaket dieses Preises dürfte es ein wenig ausführlicher sein und dem Anfänger mehr unter die Arme greifen. Vor allem ein umfangreicheres Tutorial könnte die Schulungen ersetzen, die der deutsche Distributor, die Firma DMC, nämlich noch nicht anbietet. Aber warten wir die nächste Version von DynaCADD ab, die bald den Kunden erreichen wird.

Für Profis...

... ist DynaCADD gedacht. Der Preis dürfte jeden Studenten nachhaltig schrekken. Und für die täglich mit CAD arbeitenden Zeitgenossen wie Ingenieure, Architekten usw. ist Zeit Geld. Sie brauchen Features wie den Datentausch von und nach anderen Programmen, den DynaCADD via DXF-Dateiformat erlaubt. Da allerdings DXF nur zweidimensionale Zeichnungen verwalten kann, ist der Transfer räumlicher Werke eingeschränkt. Beim Testexemplar kam hinzu, daß die DXF-Import-Funktion ihren Dienst mit einem schlichten Programmabbruch und anschließendem Totalstillstand des Rechners quittierte.

Geplant ist auch die Implementation des Programms auf verschiedenen anderen Rechnern, auf jeden Fall auf dem Apple Macintosh und unter UNIX. Um eine Aufwärtskompatibilität zu diesen Systemen (vielleicht auch nur zum ATARI TT...) zu ermöglichen, wurde der Software ein Dateiformat spendiert, das die Zeichnung in Textform beschreibt. In diesen *.DEF-Files sind alle Informationen über die Zeichnung enthalten. Wer Daten von außen in DynaCADD einbinden will (z.B. Meßergebnisse zur Darstellung als Kurven), der wird sich mit dem Datei-Format auseinandersetzen müssen.

Es ist zwar simpel, jedoch umfangreicher als z.B. die ASCII-Schnittstellen von CADproject oder TechnoCAD. Haben Sie einen PostScript-Laserdrucker, oder gar einen Satzbelichter daheim? Wenn ja, dann werden Sie die Ausgabeform 'PostScript' benutzen.

Sehr gut hat mir das Dateiprinzip von DynaCADD gefallen: Wie bei einem Projekt aktiviert man zunächst ein 'Teil' und danach eine der zugehörigen 'Zeichnungen', die in einem Ordner, der den Namen des Teils trägt, abgelegt sind. So sorgt das Programm selbstständig für Ordnung auf der Platte. Hier fehlt eigentlich nur die - für die Datensicherheit unerläßliche - automatische Speicherung in bestimmten Zeitabständen. Positiv ist auch, daß mit einer Zeichnung alle ihre Parameter inklusive Ausschnitt und Zoomfaktor gespeichert werden. Man kann nach dem Laden direkt dort weiterarbeiten, wo man das Werk verlassen hatte.

Von weitem...

... betrachtet wirkt DynaCADD wie ein behäbiger Riese. Mit seiner Programmlänge von über 500 kB, hohen Lade- und Redrawzeiten (Das Laden der Beispielzeichnung dauert immerhin 72 Sekunden, davon der Zeichnungsaufbau 25 Sekunden) scheint es kein Geschwindigkeitsweltmeister zu sein. Doch ist die Zeichnung erst einmal im Speicher, erweist sich das als falsch. Die Bearbeitungszeiten sind normal, vor allem im 2D-Teil. Daß der Redraw einer dreidimensionalen Zeichnung länger dauert, ist logisch und muß in Kauf genommen werden. Doch gibt es Funktionen wie Layer, Maskierungen oder simples Verstecken, um den Zeichnungsaufbau zu beschleunigen und die Übersichtlichkeit zu erhöhen.

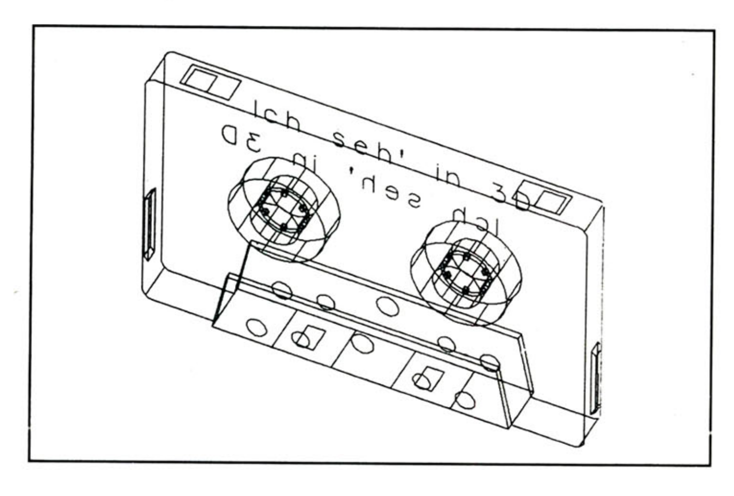


Bild 7: Beispielzeichnung (die 3D-Cassette)

In der Testversion fehlte die Schraffurfunktion noch, das Handbuch und die Hilfefunktion waren noch englisch. Das ist mittlerweile geändert. Bei DMC möchte man mit DynaCADD eine professionelle CAD-Software anbieten und sich auf den entsprechenden Markt einstellen. Dazu gehören natürlich Hotline, Schulungen etc. Man sei in diesem Bereich noch im Aufbau, war aus Walluf zu erfahren, daher habe man z.B. die bisherigen Versionen von DynaCADD auch nicht beworben. Das soll ab der jetzt bereits erhältlichen Version 1.64 anders werden.

DynaCADD hat das Zeug dazu, zu einem attraktiven Angebot im Low-Cost-CAD-Bereich zu werden. Vergleicht man die Leistung mit der entsprechender PC-Systeme und betrachtet dazu noch die erheblich preiswertere ATARI-Hardware, hat DynaCADD gute Chancen. Das Programm wendet sich damit vor allem an Anwender die ihre Probleme mit dem Werkzeug CAD lösen wollen, der ATARI ist dabei eher sekundär und eigentlich nur Mittel zum Zweck (ähnliches gilt für den Kundenkreis von TechnoCAD). Es fehlen für den Profi-Einsatz aber noch Dinge wie automatische Stücklistenerstellung, die Verfügbarkeit von Symbolbibliotheken und - solange der ST noch keine hohe Auflösung in Farbe beherrscht - die Sichtbarmachung der Linienstärken am Monitor.

Es sind Details, die den professionellen Einsatz erleichtern: Das bereits erwähnte Konzept der Dateiordnung zählt dazu. Die Arbeit läßt sich jederzeit unterbrechen und ein anderes Programm kann gestartet werden. So fungiert DynaCADD als Shell, von der aus man z.B. Zugriff auf einen Texteditor oder eine Datenbank hat.

Zu guter Letzt...

... bleibt eigentlich nur die Frage, wer den 3D-Teil, der mit Sicherheit am Umfang des Programms nicht ganz unschuldig ist, eigentlich braucht. Ich will ganz ehrlich sein: So recht fällt mir keine Anwendung an, für die das Erstellen von Drahtgittermodellen ausreichen würde und für die die Rechengeschwindigkeit, die der ST bietet, genügt. Böse Zungen behaupten ja, 3D-Programme bräuchten z.B. Architekten nur, um die Phantasielosigkeit ihrer Kundschaft auszugleichen. Doch gerade diese Anwendergruppe braucht Volumenberechnungen. Leider reichen die Fähigkeiten von DynaCADD nicht soweit.

Mit Sicherheit läßt sich in anderen Bereichen mit Hilfe von 3D-CAD-Programmen die Montierbarkeit kompletter Baugruppen bereits im Stadium der Planung gut überprüfen. Besonders im Hinblick auf Roboter-Handling und NC-Programmierung kann das wichtig sein. Doch seien an dieser Stelle wiederum Zweifel angemeldet, ob das Werkzeug DynaCADD dazu wirklich ausreicht. Es macht sicher Spaß, räumlich zu konstruieren, aber erwarten Sie nicht von einem 8 MHz-68000er, daß er Ihnen Ihre frisch entworfene Gartenlaube in Echtzeit durch den Raum trudeln läßt, wie das so manch reißerische Werbung anderer Systeme gern verspräche.

Aber allein der 2D-Teil ist es wert, näher betrachtet zu werden. Ob er 2250,- DM wert ist, das muß der Kunde entscheiden. Man wird aber - wenn DMC seine Ankündigungen hält - DynaCADD neben TechnoCAD bald in der Reihe professioneller CAD-Lösungen auf dem ST nennen müssen.

IB

DynaCADD

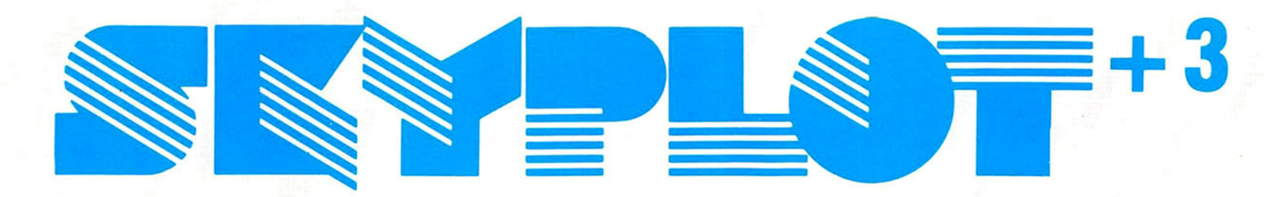
+ umfangreiche Funktionen
+ durchdachtes Konzept

- nicht absturzsicher
- hoher Preis

Bezugsadresse:

DMC
Schöne Aussicht 41
6229 Walluff
06123/71250

Preis: DM 2248.-



Ein paar Worte zum Wahnsinn:

Um es gleich deutlich zu machen: Es geht hier um das Programm **SKYPLOT**, genauer um die neueste Version **SKYPLOT PLUS 3**.

Es gab einmal einen „absoluten Wahnsinn" in Form von **SKYPLOT PLUS**, der dann dem „gesteigerten Wahnsinn" in Gestalt des Nachfolgers **SKYPLOT PLUS 2** weichen mußte. Nun fällt uns leider keine weitere Steigerung mehr zum Wahnsinn ein, wo **SKYPLOT PLUS 3** diese doch verdient hätte! Was tun?

Wir lassen also die kühlen Fakten für sich sprechen, ganz vernünftig:

„...the most sophisticated astronomical simulation package that we have ever seen on a microcomputer"
(_The_Planetarian_)

„...die Möglichkeiten sind selbst bei häufiger Benutzung kaum auszuschöpfen"
(c't)

„...gäbe es einen Oscar oder eine goldene Palme für Atari-Programme, wäre SKYPLOT ein Kandidat dafür... ein echter Grund, sich einen Atari ST zuzulegen"
(XEST)

„...ein unglaubliches Programm... einfacher und eindrucksvoller läßt sich einem Interessierten der Kosmos kaum näherbringen"
(ATARI Magazin)

„...goldenes Byte für SKYPLOT"
(Computer persönlich)

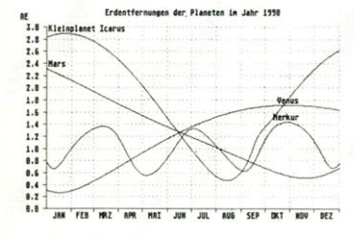
„...von dem Programm sehr begeistert... erwies sich auch im professionellen Gebrauch sehr nützlich"
(Max-Planck-Institut für Radioastronomie Bonn)

„...if you own an Atari and enjoy astronomy, you must get this program"
(Sky & Telescope)

Nun genug der Meinungen, schließlich beziehen sich die Pressestimmen ja alle auf die *alten* Versionen, denen zu **SKYPLOT PLUS 3** mindestens über ein Jahr an Entwicklungszeit fehlen!

Irgendwo muß diese Arbeit wohl stecken, und zwar hier:

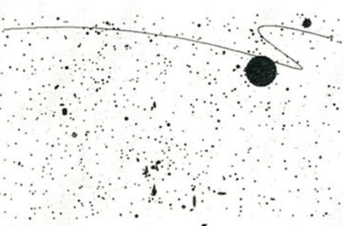
- bis zu 64000 Sterne
- bis zu 32000 Nebel etc.
- umfangreiche On-Line-Hilfen
- Echtzeitsimulation
- jede Mange Diagramme und Simulationen zur Verdeutlichung von astronomischen Sachverhalten
- Bilder laden oder speichern: IMG, AIM, STAD, Neochrome, Degas (Elite), komprimiert



- eingebaute DOS-Funktoinen: Dateien löschen, Programme starten, Formatieren
- umfangreicher Parameter für Grafik und Drucker
- Zeichnen, Speichern und Laden von Bewegungsbahnen
- Plotausgabe, auch für HPGL-kompatible Plotter
- Laden von Daten stark beschleunigt

- Finsterniskanon
- Benutzung der Hardwareuhr
- Zonenzeit und Zeitzonen
- direkte Steuerung von Vergrößerung, Ausschnitt etc. durch Tasten
- 3D-Darstellung, auch für Stereoskope
- Grafiken bearbeiten: Spiegeln, Beschriften etc.
- Speichern von Sequenzen





- eingebaute Editoren für Textdateien und Sternbildhilfslinien
- Datenausgabe auf Drucker: Ephemeriden, Kalender etc.
- jede Menge Hardcopies eingebaut, auch für 24- oder 48-Nadeldrucker
- mitgelieferter Bitmaptreiber erreicht Auflösungen von 6912★4320 Pixeln oder mehr

Schluß mit dem Wahnsinn — her mit SKYPLOT PLUS 3!

* alle Preise sind unverbindlich empfohlene Verkaufspreise

Schweiz
DataTrade AG
Langstr. 94
CH - 8021 Zürich

Österreich
Haider
Computer + Peripherie
Granzer Str. 63
A-2700 Wiener Neustadt

Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 06151-56057

BESTELL-COUPON an Heim-Verlag, Heidelberger Landstraße 194, 6100 Darmstadt-Eberstadt

Bitte senden Sie mir _____ Stück **SKYPLOT PLUS 3** für nur **DM 198.-** St.
_____ Stück **Update mit Modul** für nur **DM 80.-** St.
_____ Stück **Update ohne Modul** für nur **DM 50.-** St.

zuzügl. Versandkosten 5,– DM (unabhängig von der bestellten Stückzahl)

Name, Vorname ______________________

Straße, Hausnr. ______________________

PLZ, Ort ______________________

Benutzen Sie auch die in ST COMPUTER vorhandene Bestellkarte



Der Ritt geht weiter...

Easyrider 68000 Makro Assembler

Nachdem wir in der letzten Ausgabe den Easyrider Reassembler vorgestellt haben, wollen wir Ihnen auch den Makro-Assembler aus der Easyrider-Serie nicht vorenthalten. Es ist ein integriertes System aus tastaturorientierter Benutzeroberfläche, 2- Pass - Makro-Assembler, Linker und Bibliotheksverwalter.

Das Programm, das, wie gesagt, aus Shell, Assembler, Linker und Bibliotheksverwalter besteht, ist mit rund 50 kB angenehm kompakt. Hinzu kommt noch eine Konfigurationsdatei, die die Funktionstastenbelegung beschreibt, sowie ein kleines Utility zur Benutzung der Dateiauswahlbox innerhalb der Shell. Ein Editor wird zur Zeit noch nicht mitgeliefert, doch soll dem nach Möglichkeit abgeholfen werden. Als externer Editor bietet sich TEMPUS an, mit dem die Shell besonders gut zusammenarbeitet. Ebenfalls im Lieferumfang enthalten sind eine Include-Datei mit Makros für TOS-Funktionen und eine AES-Bibliothek einschließlich deren Quelltext. Das Handbuch ist deutschsprachig und umfaßt 51 Seiten. Easyrider ist nicht kopiergeschützt, aber auf den Anwender installiert (Name und Adresse erscheinen in der Titelzeile). Anwender, die das Programm bei einem Händler erworben haben, müssen ihre Programmdisketten mit dem Kaufbeleg dem Hersteller zusenden, um am Update-Service teilzunehmen, und erhalten dafür eine auf ihren Namen installierte Version zurück. Zum Test stand die Version 1.10 von Easyrider zur Verfügung. Kurz vor Abschluß dieses Berichts erreichte uns noch die Version 1.12, die teilweise berücksichtigt wurde. Das Programm wurde auf einem ATARI ST mit 1 MB RAM, S/W-Monitor und einer vorläufigen Version des TOS 1.4 im ROM getestet.

Bedienungskonzept

Bei der Bedienung des Assemblers wurde ein ungewöhnlicher Weg eingeschlagen. Nach dem Programmstart präsentiert sich dem Benutzer eine tastaturorientierte Shell, die keinen Gebrauch von GEM macht und nur wenige Befehle kennt, aber ein sehr effizientes Arbeiten ermöglicht. So reicht die Eingabe des Namens eines Quelltextes, um den Assembliervorgang auszulösen, wobei das Programm im allgemeinen auch gleich gelinkt wird. Durch '*RUN*: Programmname' läßt es sich anschließend starten. Bei späteren Assemblierversuchen reicht sogar ein simpler Druck auf die RETURN-Taste bzw. ein einfaches RUN:, da Easyrider sich den Namen des letzten Quelltextes sowie des letzten gestarteten Programms merkt. Falls man den Dateinamen nicht kennt, gibt es noch die Möglichkeit mittels einer Dateiauswahl-Box den gewünschten Dateinamen in die Kommandozeile zu kopieren. Ebenfalls mit dem RUN-Kommando läßt sich ein externer Editor aufrufen, um einen Quelltext erstmals einzugeben oder zu ändern. Daraus folgt schon, daß Easyrider gewöhnliche ASCII-Texte verarbeitet.

Viele Fehlerkorrekturen gestalten sich noch einfacher, da der Assembler bei jedem aufgetretenen Fehler zunächst anhält und die fehlerhafte Zeile anzeigt. Nun kann diese Zeile direkt ediert werden, wobei die Cursor-Tasten, Backspace, Delete und Clr-Home (Zeile löschen) zur Verfügung stehen. Dabei besteht die Möglichkeit, den Fehler nur für den Assemblerlauf zu korrigieren, um weitere (Folge-)Fehler zu verhindern, oder die Änderungen dauerhaft im Quelltext zu sichern. Falls die edierte Zeile nicht länger geworden ist, wird nur sie gesichert (dabei mit Leerzeichen bis zur alten Länge aufgefüllt), ansonsten wird der ganze Quelltext ab der Zeile neu gespeichert, was bei Benutzung einer RAM-

Disk aber kaum einen Unterschied macht. Falls der Fehler nicht durch Korrektur der angezeigten Zeile möglich ist (z.B. weil ein Symbol angesprochen wurde, das an anderer Stelle hätte definiert werden müssen) oder es sich um einen derjenigen (wenigen) Fehler handelt, bei denen eine Korrektur auf diese Weise nicht möglich ist, kann der Fehler übergangen werden, um ihn später (d.h. nach Ende des Assemblerlaufs) zu korrigieren. Auch hier ist Easyrider flexibel. Zum einen kann man einen externen Editor aufrufen und ihm dabei automatisch den Dateinamen und die Zeilennummer des letzten Fehlers übergeben lassen. Dies ist wohl vor allem für TEMPUS gedacht, der dann automatisch den Cursor auf die fehlerhafte Zeile positioniert. Falls einem die direkte Fehlerkorrektur nicht behagt, läßt sie sich mit einer Direktive im Quelltext auch abschalten. Dann läuft der Assembler in einem durch und bricht nur bei fatalen Fehlern ab. Alle nicht korrigierten Fehler merkt sich Easyrider. Sie können später ausgegeben werden. Nachdem der Assembler fertig ist, werden einige Angaben über Längen der Programm-Segmente, Speicherplatz- und Zeitverbrauch gemacht (s. Bild). Nun befindet man sich im sogenannten "Listing-Generator", auf den gleich noch näher eingegangen wird. Von hier aus gelangt man per Tastendruck wieder in den Kommandomodus, wo ein neuer Assembliervorgang ausgelöst oder ein externes Programm gestartet werden kann. Zur Shell wäre noch anzumerken, daß meiner Meinung nach die unterschiedliche Behandlung der RETURN- und ENTER-Tasten beim Abschließen einer Eingabezeile eher lästig als hilfreich ist. Bei RETURN werden nämlich alle Zeichen rechts vom Cursor abgeschnitten, nur bei ENTER wird die gesamte Zeile übernommen. Dies scheint mir doch recht gewöhnungsbedürftig. Ebenso muß man bei Benutzung der ESC-Taste ein wenig aufpassen, da man mit ihr stets die jeweilige Programmebene verläßt. Dabei kann es sich um harmlose Aktionen wie das Überspringen einer Fehlermeldung, den Abbruch des Assemblierens aber auch um das Verlassen des Programms handeln. Ein einmal zuviel gedrücktes ESC kann so unerwünschte Folgen haben. Ähnlich verhält es sich mit den unterschiedlichen Bedeutungen von CTRL-C.

Der Eingabe-Komfort wird noch dadurch erhöht, daß die Funktionstasten frei definierbar sind. Dabei stehen auch Platzhalter für den letzten Quelldateinamen sowie die letzte Fehlerzeile zur Verfügung. Außerdem gelangt bei SHIFT + Funktionstaste der definierte Text auch gleich zur Ausführung, ohneSHIFT wird er lediglich in die Eingabezeile kopiert. Aber es kommt noch besser, denn die letzten Eingabezeilen werden in einem 1 kB großen Puffer gehalten und können mit den Cursor-Tasten zurückgeholt werden. Dies vereinfacht die Bedienung vor allem dann, wenn mehr als ein Quelltext gleichzeitig bearbeitet wird, da dann der eine Default-Name der Shell nicht ausreicht. Als nachteilig hat sich meines Erachtens das Fehlen sämtlicher Datei-Operationen in der Shell bemerkbar gemacht. Kopieren und Löschen lassen sich noch mit TEMPUS bewerkstelligen, aber beim Umbenennen wird es schon schwierig. Hier sind entsprechende Accessories, die dann vom GEM-Editor aus aufgerufen werden können, oder eigenständige Utilities, die eigentlich zu UNIX(-like)-Shells gehören, hilfreich. Übrigens kann eine Kommandozeile Easyrider schon beim Programmstart als Parameter übergeben werden, doch bietet dies keine nennenswerten Vorteile. Easyrider sollte nämlich nicht in ein TTP-Programm umbenannt werden, da es dann Probleme beim Starten von GEM-Programmen mittels des RUN-Kommandos gibt. Die Benutzung von einem Kommandozeilen-Interpreter aus ist möglich, wird aber nicht besonders unterstützt, da man um die mehrzeile Bildschirmausgabe nach dem Assemblieren und das anschließende Prompt des Listing-Generators nicht herumkommt. Außerdem werden auch keine Standard-Suchpfade oder Environment-Variablen unterstützt. Dies erschwert demnach auch den Einsatz von Programmentwicklungssystemen wie MAKE, aber hier bietet Easyrider selbst einiges (s.u.), was zumindest bei kleineren und mittleren Programmen ausreichend sein dürfte.

```
ER 68000 Macro Assembler Version 1.10 Copyright (C) 1988 Andreas Borchard
TEST.S
Pass 1
Pass 2
0 Fehler
33 Zeilen assembliert

TEXT    DATA    BSS     SYMBOLS
8       0       0       0

40 Bytes Programmcode erzeugt
   Labeltabelle: 32
            von: 838
Arbeitsspeicher: 5305
            von: 416074
  Zeitverbrauch:.01 s
Programm listen?
```

Der Assembliervorgang ist fertig

Der Listing-Generator

Im Gegensatz zu anderen Assemblern legt man bei Easyrider erst nach dem Assemblieren fest, was man wohin gelistet haben möchte. Deswegen wird, wie oben schon erwähnt, nach dem Assemblerlauf automatisch der sogenannte Listing-Generator aktiviert. Er wird ebenfalls durch Eingabe kurzer Kommandos gesteuert, wobei auch hier ein einfaches RETURN ausreicht, ein komplettes Programmlisting auf dem Bildschirm zu erhalten. Er kann Symboltabellen, Fehlerdateien und Programmlistings ausgeben. Dabei kann die Ausgabe auf den Bildschirm, den Drucker oder in eine Datei erfolgen. Insbesondere werden hierbei mögliche Fehler (Datei schon vorhanden, Drucker nicht empfangsbereit) vernünftig behandelt. Dabei läßt sich das Ausgabeformat noch über zusätzliche im Quelltext stehende Assembler-Direktiven steuern (s.Tab. 1). Es können das Zeilen- und Seitenformat sowie die Tabulator-Breite festgelegt, Zeilen- und Seitenvorschübe eingefügt werden, dem Drucker können beliebige Steuersequenzen übergeben und eine Titelzeile kann spezifiziert werden. Weiterhin gibt es ein "Schmal-Format", bei dem der erzeugte Code nicht mit ausgegeben wird, außerdem läßt sich einstellen, was überhaupt gelistet werden soll: fehlerlose und fehlerhafte Zeilen, Warnmeldungen, Makros (als Makros oder expandiert),

Symbol-Tabelle, dabei sind verschiedene Kombinationen möglich. Das einzige, was vielleicht noch fehlt, ist eine Möglichkeit, das aktuelle Datum irgendwo in der Titelzeile unterzubringen. Nicht so ganz zufriedenstellend funktionierte die Einstellung der Tabulator-Breite. Nach dem "Komprimieren" eines vorher tabulatorfreien Textes mit TEMPUS und Einstellen der gewünschten Breite ergab sich nicht der erwartete Ausdruck. Aber da es sowieso mehrere "Tabulator-Systeme" gibt, kann man es sowieso nicht allen recht machen.

BREIT	ERRS	EXP
LCHAR	LIST	LLEN
NOLIST	NOEXP	PAGE
PLEN	SCHMAL	SPACE
SYMB	TAB	TTL

Tab. 1: Direktiven des Listing-Generators

Der Assembler

Kommen wir nun zum eigentlichen Assembler. Hier zeigt sich Easyrider von einer etwas eigensinnigen Seite, was das Format des Quelltextes angeht. Die Probleme ergeben sich im wesentlichen dadurch, daß Leerzeichen strikt als Trennzeichen zwischen den einzelnen Feldern einer Zeile (Marke, Mnemonic, Operanden, Kommentar) angesehen werden. Marken können mit einem Doppelpunkt beendet werden, jedoch dürfen zwischen der Marke und dem Doppelpunkt keine Leerzeichen stehen, was aber ein beliebtes Mittel zur Verbesserung der Lesbarkeit von Kolonnen von Symbol-Definitionen ist. Ebenso können auch Kommentare durch Leerzeichen abgetrennt werden (neben den üblichen Trennzeichen Stern und Semikolon), was zu fatalen Fehlern führen kann. So wird z.B. *move.l d0,symbol +4(a0)* wegen des Leerzeichens vor dem + zu *move.l d0,symbol* assembliert, da *+4(a0)* als "Kommentar" interpretiert wird. Des weiteren dürfen Direktiven nicht mit einem Punkt beginnen, so daß man hier erst einen Editor bemühen muß, um seine alten Quelltexte anzupassen. Nicht nur, daß dies zur Schwierigkeiten bei der Portierung bereits vorhandener Quelltexte führt, sondern es sind auch leicht Fehler bei der Eingabe neuer möglich. Der Assembler ist auch nicht in der Lage, *tst.l (a0,d0)* zu *tst.l 0(a0,d0)* zu ergänzen, wobei dies allerdings eher als Kleinigkeit anzusehen ist. Ansonsten kann man sich aber nicht über die Syntax beklagen. Es werden selbstverständlich alle 68000-Adressierungsarten und Registernamen (einschl. SP,CCR,SR,USP) in der von Motorola festgelegten Syntax akzeptiert. Ebenso ist die Endung .S oder .W zum Festlegen der "Short-Adressierung" zugelassen.

Auch bei den Operatoren in Ausdrücken findet man alles, was man braucht: Grundrechenarten, bitweises AND, OR, XOR, NOT sowie die unären Vorzeichenoperatoren. Die Operatoren sind priorisiert, zusätzlich sind Klammern erlaubt. Es wird grundsätzlich mit 32-Bit-Arithmetik gerechnet. Desweiteren kann auch auf den aktuellen Programmzähler zugegriffen werden. Der Assembler arbeitet im allgemeine tadellos, hat aber anscheinend zumindest einen Fehler. Wenn 'marke1' und 'marke2' im Programmtext definierte Marken sind, führt

```
LABEL = marke1 - marke2
moveq #LABEL,d0
```

zu einer Fehlermeldung, aus der hervorgeht, daß der Assembler LABEL für ein zu relozierendes Symbol hält, was aber nicht der Fall ist. Dagegen funktioniert

```
moveq #marke1 - marke2
```

einwandfrei. Aber damit läßt sich noch leben, denke ich.

Direktiven des Assemblers

Ein Assembler soll bekanntlich erlauben, ein Programm auf möglichst bequeme Weise in Assemblersprache zu formulieren. Dazu dienen unter anderem die Direktiven, die ähnlich wie die Mnemonics der eigentlichen Assemblersprache gehandhabt werden Ich möchte hier nur auf die Besonderheiten eingehen, die nicht selbstverständlich sind oder die hier anders als üblich realisiert sind. Eine Übersicht aller Direktiven, die für Assembler (und Linker) relevant sind, zeigt die Tab.2. Easyrider erlaubt die Benutzung lokaler Marken. Sie beginnen mit einem Punkt und ihr Gültigkeitsbereich reicht bis zur nächsten LOCAL-Direktive. Dies ist sicherlich flexibler als die Methode anderer Assembler, bei denen lokale Marken automatisch bei der nächsten globalen Marke ungültig werden, hat aber den Nachteil, daß man - bei extensiver Nutzung lokaler Marken - praktisch jedes Unterprogramm mit einer LOCAL-Direktive beginnen muß. Mit SET definierte Marken dürfen jederzeit umdefiniert werden, im Gegensatz zu den mit EQU oder = definierten. EQUR erlaubt die Benennung von Registern, so daß Register mit symbolischen Bezeichnungen angesprochen werden können. Diese Definitionen lassen sich auch durch die Symbol-Tabelle exportieren. Registerlisten für den MOVEM-Befehl werden nicht unterstützt. ALIGN läßt sich auch zum Justieren des Programmzählers auf Langwort-Adressen verwenden. Da Assemblerprogramme häufig als Ergänzungen zu Hochsprachen-Programmen geschrieben werden, halte ich es für wichtig, daß ein Assembler das Arbeiten mit strukturierten Variablen-Typen erleichtet ('structures' bzw. 'records' für die C- bzw. Pascal-Freunde). Leider bietet Easyrider keine Möglichkeit, solche Strukturen einfach zu definieren (wie etwa mit der ABS-Direktive des MADMAC-Assemblers), sondern es bleibt nur die herkömmliche Methode, alle Komponenten einzeln mit 'EQU' festzulegen. Ebenfalls für wichtig halte ich die Möglichkeiten eines Assemblers, relative Programmierung zu unterstützen, da dies im allgemeinen Programmcode und Ausführungszeit einsparen hilft. Neben der Code-Optimierung, die unten noch erläutert wird, bietet Easyrider hier die interessante BASE-Direktive. Betrachten wir dazu ein kleines Beispiel:

```
         text
         lea   base5,a5
         clr.l var2(a5)
         bss
base5:
var1:    ds.w 1
var2:    ds.l 1
```

base5 ist dabei zunächst eine normale Marke, ist aber dem Adressregister 5 in der Weise zugeordnet, daß der 'clr'-Befehl wie ein *clr.l var2-base5(a5)* assembliert wird. Der integrierte Linker bietet

ALIGN	BASEn	BSS
DATA	DC	DCB
DS	END	EQU, =
EQUR	EVEN	GLOBL
IF..ELSE..ENDIF	INCLUDE..END	LOCAL
MACRO..ENDM	OPT	ORG
REPEAT..UNTIL	SECTION	SET
TEXT	XDEF	XREF

Tab. 2: Assembler-Direktiven

noch die Besonderheit, daß er relative Zugriffe (im assemblierten Quelltext!) z.B. vom TEXT ins DATA- oder BSS-Segment zuläßt. Somit ist es also möglich, relativ zu programmieren und trotzdem die saubere Trennung zwischen Programm und Daten beizubehalten. Bei der bedingten Assemblierung mittels IF..ELSE..ENDIF können zwei Ausdrücke miteinander oder ein Ausdruck mit Null verglichen werden, wobei die möglichen Bedingungen EQ, NE, GE, LE, GT und LT sind. Die Ausdrücke dürfen hier sogar Zeichenketten sein. Auch möglich sind Abfragen, ob ein Symbol definiert ist oder ob nicht. Witzig ist die Möglichkeit, quasi eigene Fehlermeldungen zu konstruieren. Z.B. wird bei

```
IFEQ A,B
     DIES IST EIN FEHLER
```

der Text "Unbekanntes Befehlswort DIES IST EIN FEHLER" ausgegeben, wenn die Bedingung A=B erfüllt ist. REPEAT..UNTIL arbeitet ähnlich wie IF und dient der Assemblierung sich wiederholender Befehle. Die maximale Verschachtelungstiefe liegt bei 4. Ferner können hiermit Schleifen mit einer festen Zahl von Durchläufen realisiert werden. Kommen wir nun zu den Makros, einem weiteren wichtigen Punkt eines jeden Makro-Assemblers. Es können maximal 99 Parameter übergeben werden, wobei die Parameter durch \1 bis \99 angesprochen werden. Außerdem kann auf die Extension mittels \0 zugegriffen werden. Der Spezial-Parameter \@ expandiert zur laufenden Nummer des Makro-Aufrufs, um für jede Makro-Expansion individuelle Marken generieren zu können. Es ist zwar nicht direkt vorgesehen, die Existenz eines Parameters abzufragen, was durchaus nützlich sein kann, aber auf Grund der Arbeitsweise der Ausdrucks-Auswertung ist dies mit einem kleinen Trick (auf den man natürlich erstmal kommen muß) trotzdem möglich. Die Include-Direktive dient neben dem Einfügen von weiteren Quelltexten (Verschachtelung laut Handbuch bis 255 Ebenen tief möglich) auch der Einbindung von Objekt-Dateien und Bibliotheks-Dateien, worauf noch bei der Beschreibung des Linkvorgangs eingegangen wird. Beim Einfügen von Quelltexten wartet Easyrider noch mit einer weiteren Besonderheit auf. Normalerweise lädt Easyrider Include-Dateien speicherresident, d.h. sie werden über beide Durchläufe des Assemblers im Speicher gehalten, sodaß der Assembler effektiv (was die Massenspeicher-Zugriffe angeht) zum 1-Pass-Assembler wird. Dies kann bei mehreren langen Include-Dateien naturgemäß zu Speicherplatzproblemen führen. Daher bietet der Include-Befehl die Option, die einzufügende Datei bei jedem Pass neu lesen zu lassen. Dies ist aber mit der Einschränkung verbunden, daß dann keine Segmentwechsel und Makrodefinitionen in den Include-Dateien erlaubt sind. Dies wird aber wohl nur bei sehr großen Programmen und wenig Speicher notwendig werden, so daß dies keine wesentliche Beschränkung bedeutet.

Code-Optimierung

Nur wenige Assembler nutzen die vielfältigen Möglichkeiten des 68000-Befehlssatzes, den erzeugten Code zu optimieren. Easyrider bildet da leider keine Ausnahme. Die OPT-Direktive erlaubt es, drei Optimierungsmöglichkeiten getrennt ein- und auszuschalten. Dabei handelt es sich um die automatische Umwandlung von "langen Sprüngen" (Bcc.W) zu "kurzen" (Bcc.S), die Änderung der absoluten Adressierungsarten in die entsprechenden Programmzähler-relativen und der Optimierung der "absolut lang"- zur "absolut kurz"-Adressierung. In den beiden letzten Fällen gilt, daß die Optimierung durch explizite Angabe der Operandengröße (.W bzw .L) verhindert werden kann. Außerdem findet nur Rückwärtsoptimierung statt. Alle anderen Codeverbesserungen einschließlich der Vorwärtsoptimierung bei Verzweigungen muß man also entweder per Hand machen oder bleiben lassen, je nach persönlichem Geschmack. Immerhin werden bei möglichen Vorwärtsoptimierungen Warnmeldungen generiert, die man sich vom Listing-Generator ausgeben lassen kann.

Linken

Normalerweise werden direkt ausführbare Programme erzeugt; mittels einer Option in der Kommandozeile, die den Assembliervorgang startet, lassen sich jedoch auch Objekt-Dateien im DR-Format erzeugen. Ebenso läßt sich auf diese Weise verhindern, daß die Objekt-Dateien bzw. Programme überhaupt gespeichert werden. Sollen Objekt-Dateien mit eingebunden werden, so müssen diese im Quelltext mit der 'include'-Direktive spezifiziert werden. Eine Angabe über die Kommandozeile ist nicht möglich, daher ist ein wirklich eigenständiges Linken ohne Assemblieren also nicht vorgesehen. Der Linker verarbeitet dabei neben dem DR- auch das GST-Format, so daß das Einbinden fremder Objekt-Dateien möglich ist. Leider hält sich der Assembler beim Erzeugen von Objekt-Dateien nicht ganz an das DR-Objekt-Format. Es wird nämlich nicht für jedes Code-Wort ein Relozierwort erzeugt, sondern am Ende werden Null-Worte weggelassen. Dadurch ist die Relozierdatei im allgemeinen zu kurz. Daher verweigert z.B. TURBO C das Linken einer solchen Datei. Der Linker von LASER C ist da weniger mißtrauisch und ergänzt die fehlenden Daten korrekt mit Null. Für das Assemblieren zu Objekt-Dateien gibt es noch die Direktiven XDEF und XREF. Mit XDEF lassen sich Marken und absolute Bezeichner über die assemblierte Datei hinaus global machen, d.h. andere Objekt-Dateien haben auf Marken nur Zugriff, wenn sie mit XDEF exportiert werden. Anders ausgedrückt, werden mit XDEF deklarierte Marken in die Symbol-Tabelle der Objekt-Datei aufgenommen. Alternativ hierzu können mit einer Form der OPT-Direktive auch alle globalen, relativen Marken pauschal in die Symbol-Tabelle übernommen werden. XREF ist das Gegenstück zu XDEF und erlaubt das Importieren externer Referenzen.

Bibliotheken

Des weiteren ist die Benutzung von DR-Bibliotheken möglich; diese setzen sich aus einzelnen Objekt-Dateien zusammen, von denen nur die tatsächlich benötigten in das Programm eingebunden werden. Dazu existiert ein Bibliotheksverwalter, der ebenfalls direkt in Easyrider integriert ist. Er läßt sich aus der Shell heraus aufrufen und ermöglicht das Anlegen, Löschen, Extrahieren und Überschreiben von Objekt-Dateien ("Moduln") in einer Bibliotheks-Datei. Außerdem kann man sich die Symbol-Tabelle einer Objekt-Datei anzeigen lassen. Dies geschieht menu-gesteuert, d.h., die Objekt-Dateien der Bibliotheks-Datei werden aufgelistet und können mit den Cursor-Tasten ausgewählt werden, und die einzelnen Funktionen können mit den Funktionstasten ausgelöst werden. Das Einfügen und Extrahieren läßt sich auch direkt von der Easyrider-Shell aus bewerkstelligen. Mittels der MODUL-Direktive lassen sich alle Quelltexte einer Bibliothek in einem Quelltext zusammenfassen. Dieser läßt sich mit einem Kommando in die einzelnen Module übersetzen, wobei die Module dann auch gleich automatisch zu einer

Bibliotheks-Datei zusammengesetzt werden. Auch die benötigten Bibliotheks-Dateien müssen im Quelltext mit 'include' angegeben werden. Da der Assembler beim 'include' schon wissen muß, welche Routinen der Bibliothek benötigt werden, muß das 'include' allerdings am Ende stehen, was nicht gerade die Übersichtlichkeit des Quelltextes fördert. Abhilfe schafft eine Voraus-Deklaration im Quelltext, was aber wiederum zu mehr Tipparbeit führt. Hier wäre das konventionelle Verfahren, Bibliotheken dem Linker als Parameter zu übergeben, doch vorteilhafter. Es ist ebenfalls möglich, absoluten Code zu erzeugen, d.h. ein reines 68000-Programm ohne GEMDOS-Programm-Kopf und Relozierdaten. Dazu dient die ORG-Direktive, mit der die absolute Adresse, für die das Programm assembliert werden soll, festgelegt wird.

TOS-Bindings

Beim Schreiben längerer Assembler-Programme wird man zwangsläufig auf die Unterstützung durch das TOS angewiesen sein. Daher stellt sich die Frage, wie komfortabel TOS-Aufrufe möglich sind. Easyrider bietet eine Include-Datei mit Makros für die BIOS-, XBIOS- und GEMDOS-Funktionen. Diese sind nicht besonders raffiniert (Parameter-Übergabe fast immer mit MOVE, manchmal PEA), aber bei der Stack-Korrektur wird immerhin Gebrauch vom ADDQ-Befehl gemacht, wo es möglich ist. Unverständlicherweise fehlen außerdem einige Funktionen (Bioskeys, Prtblk, Blitmode, Super, Tget..., Tset..., Sversion). Weiterhin gibt es eine AES-Library, die bis auf die neuen Funktionen des TOS 1.4 vollständig ist. Die Bindings sind allerdings sehr umständlich programmiert (sieht aus wie schlecht kompiliertes C). Unterstützung für VDI, LINE A und die Systemvariablen fehlt völlig. Hier gäbe es also noch einiges zu verbessern.

Geschwindigkeit

Ein weiteres Kriterium zur Beurteilung eines Assemblers ist dessen Arbeitsgeschwindigkeit. Auf aufwendige Benchmarks habe ich verzichtet, da ich deren Aussagekraft hier für zweifelhaft halte. Easyrider ist aber auf jeden Fall recht schnell, fixer noch als der MADMAC-Assembler von Atari. Für meinen Geschmack ist er schnell genug, unangenehme Kaffeepausen entstehen nicht. Um Ihnen dennoch einen genaueren zeitlichen Rahmen zu geben, habe ich einen ca. 50 kB langen Quelltext mit einer 15 bzw. 22 kB (MADMAC bzw. Easyrider) großen Include-Datei assembliert. Auf RAM-Disk benötigte Easyrider 2,9 s, MADMAC dagegen 3,9 s. Bei Diskette lagen die Zeiten bei 13,4 s bzw. 14,6 s. Dabei handelt es sich nur um die Assemblier- und Linkzeiten, die Ladezeit des Assemblers bleibt unberücksichtigt. Die Include-Datei wurde dabei speicherresident geladen. Wie man an den Disketten-Zeiten sieht, ist Easyrider damit quasi ein 1-Pass-Assembler, auch wenn es intern zwei Durchläufe sind. Da Easyrider im allgemeinen nur einmal geladen werden muß, vergrößert sich sein Vorsprung damit nochmals. Ebenso spart das direkte Korrigieren vieler Fehler oftmals das Laden des Editors, was die Turn-around-Zeit weiter senkt. Bei dieser Gelegenheit sollte auch einmal die Betriebssicherheit hervorgehoben werden. Bei mir gab es in der Testphase keinen einzigen Absturz oder irgendwelche anderen Ungereimtheiten, die keine "natürliche" Ursache hatten.

Handbuch

Das Handbuch bietet einen Schnellkursus für Lesefaule, der dem erfahrenen Programmierer ein sofortiges Arbeiten ermöglicht. Die Anleitung bietet alle wesentlichen Informationen und läßt kaum Fragen offen, die erst durch Ausprobieren beantwortet werden müssen. Die Formulierungen sind knapp aber präzise. Beispiele ergänzen die Erklärung. Wichtige Passagen sind durch Fettdruck hervorgehoben. Ebenso werden Spezialfälle und Klippen ausreichend behandelt. Vorausgesetzt werden Kenntnisse des 68000-Assemblers, da der Befehlssatz und die Adressierungsarten nur tabellarisch aufgelistet sind. Einzig die Gliederung ist manchmal etwas merkwürdig, so sind die Möglichkeiten der Shell über mehrere Kapitel verstreut. Hier wäre eine Zusammenfassung der Kommandozeilen-Syntax, Assembler-Aufrufmöglichkeiten und Edierfunktionen angebracht. Die Liste der Fehlermeldungen ist unsortiert, so daß man ein wenig suchen muß. Außerdem wäre es für Anfänger sicher hilfreich, wenn hier bei allen Meldungen einige Tips zur Behebung des Fehlers stünden.

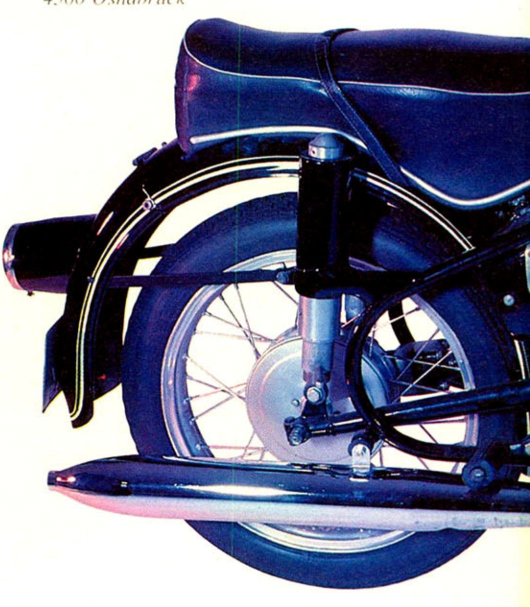
Fazit

Zusammengenommen läßt sich urteilen, daß man mit Easyrider ein leicht und effizient zu bedienendes Assembler-System in der Hand hält. Der Assembler ist allerdings beim Format der Quelltexte zu pingelig oder großzügig (wie man's nimmt), was das Portieren vorhandener Quelltexte erschwert und Fehler begünstigt. Der Assembler bietet die wichtigsten Direktiven und ein paar komfortable darüber hinaus (BASEn, EQUR, LOCAL), erfüllt aber nicht alle Wünsche, die man als Assembler- und Hochsprachen-Programmierer so hat (aber welcher ST-Assembler tut das schon - ich kenne jedenfalls keinen). Die Ausgabemöglichkeiten sind vielseitig, das ganze System ist schnell und betriebssicher. Modulare und relative Programmierung werden gut unterstützt. Der Linker und insbesondere die TOS-Unterstützung sind noch verbesserungsbedürftig. Auch die Code-Optimierungen sind nicht gerade berauschend. Das Handbuch halte ich für gut und vollkommen ausreichend. Insgesamt ragt Easyrider zwar nicht besonders aus allen anderen Assemblern heraus, kann aber durchaus mit anderen Systemen mithalten und sollte daher einer näheren Betrachtung wert sein.

Alex Esser

Easyrider Assembler
Einzelpreis DM 59,-
Paketpreis (mit Reassembler) DM 198,-
Update-Gebühr DM 10,- für registrierte User

Bezugsadresse:
Andreas Borchert
Wiesenbachstr. 2a
4500 Osnabrück



FRANKIE KILLER

Reiner Tisch unter Aladin

Kennen Sie schon Frankie? Ach, Sie arbeiten nicht mit Aladin? Doch? Na, dann werden Sie ihn bestimmt noch kennenlernen. Ich habe Frankies Bekanntschaft vor etwa einem Monat gemacht. Zur Zeit bin ich dabei, meine Studienarbeit auf dem ATARI zu schreiben. Und da ich neben der reinen Textverarbeitung auch noch Bilder und Diagramme malen muß, habe ich mich entschlossen, das ganze auf dem ATARI, aber unter dem Aladin-Betriebssystem zu machen. Dort habe ich dann mit den Programmen MacWrite und MacPaint gearbeitet.

Am Anfang ging auch alles gut, bis eines Tages dann Frankie tauchte: Ich war gerade an einer Zeichnung, als sich plötzlich die gesamte Menü-Leiste nach unten verschob und hinter meiner Zeichnung verschwand. Direkt danach tauchte sie wieder auf, aber diesmal stand da nicht 'Ablage Bearbeiten usw..', sondern 'Frankie says: No more Piracy', und ab diesem Augenblick tat sich überhaupt nichts mehr. Die Maus bewegte sich nicht mehr, es ließen sich keine Disketten mehr einlegen oder auswerfen und selbst der Griff zum Reset-Knopf, der bisher bei jedem 'schwerwiegenden Systemfehler' geholfen hat, brachte mich nur zurück zum schnöden ATARI-Desktop.

Was tun? Ich habe (dummerweise) gleich darauf untersucht, ob meine anderen Mac-Programme noch laufen. Bis ich auf die Idee kam, daß sie dadurch vielleicht auch infiziert werden, hatte ich schon mindestens 3/4 aller meiner Programme getestet und natürlich auch infiziert. Das Gemeine an Frankie ist ja, daß er den Benutzer am Anfang ganz normal arbeiten läßt, bis er dann nach knapp zehn Minuten in Erscheinung tritt. Und wenn er erst einmal aufgetreten ist, dann läßt sich das bisher Erstellte auch nicht mehr abspeichern.

Da Frankie immer knapp zehn Minuten nach dem Starten eines Programmes auftaucht, vermutete ich, daß er sich vielleicht nach der internen Uhr richtet. Deswegen versuchte ich alles mögliche, um sein Auftreten zu verhindern: Uhrzeit vor dem Programmstart zurückstellen, Uhrzeit direkt nach dem Aufruf des Programmes zurückstellen, Uhrzeit mehrfach verstellen, aber leider alles ohne Erfolg.

Dann habe ich versucht herauszufinden, wo sich Frankie versteckt. Schon bald hatte ich bemerkt, daß die infizierten Programme plötzlich etwa 3 kB größer waren als ursprünglich. Ich stellte fest, daß Frankie die Programme bereits beim Kopieren von einer Diskette auf eine andere verändert! Da mir aber nach einiger Zeit die Lust verging, immer nach 10 Minuten das ganze System wieder neu zu starten, beschloß ich, das Virusproblem einmal grundlegend zu untersuchen. Ich besorgte mir ein sauberes System: *ResEdit* (ein Resource-Editor von Apple) und *MacTools* (ein Diskettenmonitor).

Grundlagen

Zunächst einmal ein paar grundlegende Fakten zum Apple. Eine Datei besteht beim Macintosh immer aus zwei Teilen: dem Datenteil und dem Resourceteil. Im Datenteil werden anwenderspezifische Daten gespeichert, wie z.B. der Text, den ich mit einem Textverarbeitungsprogramm erstellt habe. Im Resourceteil werden Informationen über Dialogboxen, Alarmboxen usw. wie man sie z.B. auch vom ATARI und dessen Resourcedateien kennt, gespeichert, aber auch der Programmcode selbst. Dies ist auch ein Grund dafür, daß Macintosh-Programme in der Regel nur aus einer einzigen Datei bestehen: Ein Programm speichert die unzähligen Parameterdateien, die man aus der IBM-Welt kennt, im Daten-oder im Resourceteil der Programmdatei ab und kommt deswegen mit einer einzigen Datei aus.

Es müssen aber nicht jedesmal beide Teile einer Datei existieren. Bei einer Datei, in der z.B. meine Studienarbeit abgespeichert ist, existiert kein Resourceteil. Zu meiner anfänglichen Verwirrung trug auch bei, daß ein und diesselbe Datei auf verschiedenen Medien (Laufwerken) unterschiedlich groß sein kann. Kopieren Sie einmal eine Datei von einer Diskette auf die SuperDisk. Sie werden in den meisten Fällen feststellen, daß die Datei angeblich kleiner wird. Das liegt daran, daß im Directory des Macintoshs nur die Größe der Datei in BLOCKs angegeben wird. Ein BLOCK entspricht den Clustern beim ATARI. Offensichtlich wird bei der Superdisk mit kleineren BLOCKs gearbeitet. Die Standardgröße eines BLOCKs beträgt bei den Aladin-Disketten 1536 Bytes (3*512). Wenn also eine Datei 800 Bytes groß ist, erscheint im normalen Directory eine Größe von 1,5 kb. Die wahre Größe einer Datei erfährt man nur über Propeller-I (Propeller ist die sogenannte Befehlstaste des Macs, bei Aladin die Control-Taste).

Aber jetzt wieder zu Frankie. Wie bereits gesagt, wird in der Resourcedatei jedes Programmes im Unterschied zum ATARI auch der eigentliche Programmcode abgespeichert. Mit Hilfe des Resource-Editors *ResEdit* kann man sich die Resourcen eines beliebigen Programmes auflisten lassen (s. Bild 1). Es gibt noch einen anderen Resource-Editor, *REdit*, aber der kann nur bestimmte Resource-Typen verarbeiten, und ausgerechnet der 'Code'-Typ gehört nicht dazu. Resourcen eines bestimmten Typs werden zusammengefaßt und erhalten einen Namen, der immer aus vier Großbuchstaben besteht. Die Kombinationen aus Kleinbuchstaben sind von Apple für eigene Zwecke reserviert. Andere Resource-Typen und -Namen sind bereits von Apple vordefiniert;

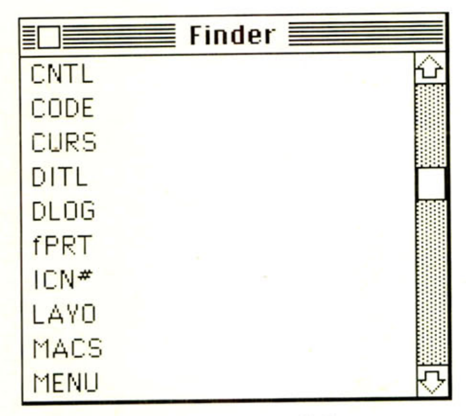


Bild 1: Mit ResEdit lassen sich die Resourcen von Programmen auflisten

ein Programmierer kann jedoch auch seine eigenen Namen für seine Resourcen verwenden.

Unter dem Namen *'ALRT'* sind z.B. alle Alert- (Alarm-) Boxen zusammengefaßt; die Icons, die ein Programm verwendet, stehen unter *'ICN#'* oder *'ICON'* (s. Bild 1). Uns interessieren jetzt aber nur die *'CODE'*-Resourcen. Unter diesem Namen verbirgt sich der eigentliche Programmcode. Und auch hier kann der Macintosh mit einigen Besonderheiten glänzen. Der Programmcode kann vom Programmierer in verschiedene Segmente aufgeteilt werden. Über verschiedene Flags, die der Programmierer vergibt, können die Eigenschaften der Segmente beeinflußt werden (diese Flags können für alle Resourcen vergeben werden). So kann das Macintosh-Betriebssystem angewiesen werden, bestimmte Code-Segmente bei zu geringem Speicherplatz aus dem Speicher zu löschen und erst wieder bei Bedarf von Diskette oder Festplatte zu laden.

Mit Hilfe des Programmes *ResEdit* kann der (erfahrene) Anwender fast jedes Programm nach seinen Wünschen verändern. Nehmen wir an, Sie haben ein Programm erstellt und wollen es in Amerika verkaufen. Da bei einem richtigen Maccie-Programm der gesamte Dialog mit dem Benutzer über Dialogboxen und Menüs abläuft, brauchen Sie bloß die entsprechenden Stellen in der Resource-Datei mit *ResEdit* zu ändern, das Programm wird davon nichts merken.

Und gerade diese Offenheit macht sich der Frankie-Virus zunutze. Er kopiert seinen eigenen Programm-Code einfach als zusätzliches Code-Segment in die Resource-Datei eines Programmes. Da der Macintosh schon während des normalen Betriebes ständig auf die System- und Programmdateien zugreift, wird man von einem solchen Kopiervorgang in der Regel nichts merken. Haben Sie schon einmal 'aus Versehen' ein Programm von der Diskette gestartet, wenn Sie nur mit einem Laufwerk arbeiten und auch das System nicht auf der SuperDisk installiert ist? Vielleicht können Sie sich jetzt besser vorstellen, wieso man in einem solchen Fall ständig die Disketten wechseln muß.

Wie findet man nun den Virus?

Als erstes öffnet man bei einem infizierten Programm den Code-Eintrag in Bild 1 durch Doppelklick (in Bild 1 wird gerade der Finder untersucht). Dann sollte so etwas ähnliches wie in Bild 2 erscheinen. In diesem Finder gibt es also sechs verschiedene Code-Segmente, würde man jetzt vermuten. Fast richtig, aber Segment Null hat eine besondere Funktion. Im Segment Null jedes Programmes wird eine Sprungtabelle abgespeichert. Da ein Programmierer beim Erstellen des Programmes nicht wissen kann, an welcher Stelle im Speicher später einmal die verschiedenen Code-Segmente stehen, legt er eine Sprungtabelle an, in der für jeden (Programm-)Sprung von einem Segment zum anderen ein Eintrag steht. Und auch hier schlägt der Virus zu.

Der Virus untersucht bei einem gestarteten oder zu kopierenden Programm die Code-Tabelle. Falls das Programm noch nicht infiziert ist, sucht er in der Code-Tabelle die höchste Nummer. Daraufhin legt er ein neues Code-Segment mit einer um Eins höheren Nummer an. In dieses Segment kopiert er dann den eigenen Programm-Code. Das allein aber reicht noch nicht, denn irgendwann soll sein Programm-Code ja auch einmal ausgeführt werden. Deshalb schreibt der Virus an die

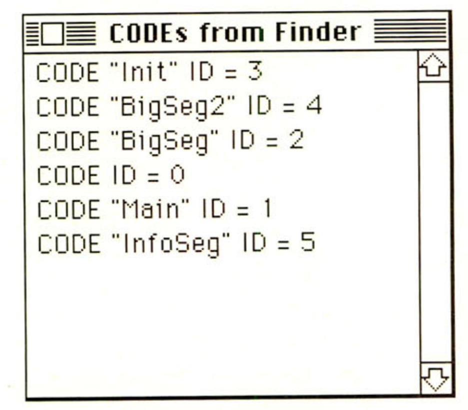


Bild 2: Der Code-Eintrag unseres Beispiels

erste Stelle in der Sprungtabelle einen Sprung ins eigene Programmsegment und, das ist wichtig, rettet den alten Eintrag im eigenen Programm-Code. Dadurch wird nach dem Start eines infizierten Programmes immer zuerst das Virusprogramm ausgeführt und dann mit Hilfe des geretteten alten Eintrags aus der

```
CODE ID = 0 from Finder
000000    0000 0CF8 0000 05C2   □□□□□□□¬
000008    0000 0CD8 0000 0020   □□□ÿ□□□
000010    0008 3F3C 0006 A9F0   □□?<□□©□
000018    0140 3F3C 0001 A9F0   □@?<□□©□
000020    017E 3F3C 0001 A9F0   □~?<□□©□
000028    0594 3F3C 0001 A9F0   □î?<□□©□
000030    0A0A 3F3C 0001 A9F0   □□?<□□©□
000038    0A74 3F3C 0001 A9F0   □t?<□□©□
000040    0CEC 3F3C 0001 A9F0   □□?<□□©□
000048    0DA0 3F3C 0001 A9F0   □†?<□□©□
000050    0E7C 3F3C 0001 A9F0   □|?<□□©□
000058    0ECE 3F3C 0001 A9F0   □Œ?<□□©□
000060    0F6A 3F3C 0001 A9F0   □j?<□□©□
000068    1036 3F3C 0001 A9F0   □6?<□□©□
```

Bild 3: Frankie erkennt man an der dritten Zeile

Sprungtabelle mit dem ursprünglichen Programm fortgefahren. Dem Benutzer bleibt dies verborgen.

Den Frankie-Virus erkennt man am Eintrag *'0008 3F3C XXXX A9F0'* an der ersten Stelle in der Sprungtabelle (Code 0) (s. dritte Zeile in Bild 3), wobei an der Stelle von XXXX dann die Nummer des neuen Code-Segments steht. Die ersten 16 Bytes der Sprungtabelle enthalten unter anderem Informationen über die Größe und Organisation der Sprungtabelle. Das Virussegment selbst ist zwischen

3400 und 3800 Bytes groß. Die Größe eines bestimmten Eintrags erfährt man über die 'Informationen'-Funktion von *ResEdit*. Aber Vorsicht: Es gibt auch andere, saubere Programme, deren erster Eintrag '0008' lautet. Meistens verrät sich der Virus aber durch die Tatsache, daß 'normale' Programme an der ersten Stelle nie einen Sprung in das Segment mit der höchsten Nummer haben.

Noch ein weiterer Tip: Gelegentlich kommt es vor, daß Programme Code-Segmente mit den Nummern 0 bis 10 und 15 bis 20 haben (Zahlen sind willkürlich gewählt). In diesem Fall steckt der Virus im Segment 10.

```
virus
  Buffer: 0000
  Sector: 0001
    Tags: Not supported on this volume
000200: 0110 598F 2F0A 3F3C 0013 4EAD 0662 201F   ..Y./.?<..N≠.b .
000210: 2F00 486E FEFA A990 486E FEFA 2F0B 4EAD   /.Hn..϶.Hn../.N≠
000220: 02A2 2F0A 3F3C 0015 486E FFFE 486E FFFA   .¢/.?<..Hn..Hn..
000230: 486E FEE0 A98D 0C6E 0010 FFFE 6600 00D2   Hn..϶..n....f.."
000240: 2F0B 4EAD 0A92 4A80 588F 6700 00C4 598F   /.N≠..J.X.g..ƒY.
000250: 2F0A 3F3C 0015 4EAD 0662 201F 2F00 486E   /.?<..N≠.b ./.Hn
000260: FEFA A990 598F 486E FEFA 4EAD 086A 201F   ..϶.Y.Hn..N≠.j .
000270: 2600 700A E1A3 2F0B 486E FEE8 4EAD 0AB2   &.p..È/.Hn..N≠.≤
000280: 3000 508F 6600 007A 3F2E FEEA A998 4A83   0.P.f..z?...϶.J.
000290: 4E75 0000 0000 0000 0000 0000 FFF8 0000   Nu..............
0002A0: 0000 0000 3F3C 0001 A9F0 7000 2078 02AE   ....?<..϶.p. x.Æ
0002B0: 558F 2053 4868 0062 3F3C 1401 4EAD 06AA   U. SHh.b?<..N≠.™
0002C0: 7000 101F 1000 671A 2D43 FEF2 6014 2F3C   p.....g.-C..`./<
0002D0: FFFF FFFF 486E FEF0 2F0B 4EAD 0A9A 4FEF   ....Hn../.N≠..O.
0002E0: 000C 486E FEF0 2F0B 4EAD 0AA2 3F2E FEE8   ..Hn../.N≠.¢?...
0002F0: A998 2F0B 486E FEE8 4EAD 0ABA 4FEF 0010   ϶./.Hn..N≠.∫O...
000300: 4A40 670C 3F3C 1414 3F00 2F0B 4EAD 069A   J@g.?<..?./.N≠..
000310: 4CEE 0C0C FED0 4E5E 4E75 2F0A 242F 0008   L....-N^Nu/.$/..
000320: 2F02 4EAD 097A 42A7 2F02 45ED 0C12 2F0A   /.N≠.zBß/.E.../.
000330: 4EAD 0A42 4FEF 0010 245F 205F 588F 4ED0   N≠.BO...$_ _X.N-
```

Bild 4: Zum Entfernen des Virus ist ein Diskettenmonitor notwendig.

Wie beseitigt man den Virus?

Jetzt wird es ja eigentlich erst richtig interessant, denn was nützt einem die Kenntnis, daß und wo ein Virus steckt, wenn man ihn nicht auch beseitigen kann. Man startet *ResEdit*, öffnet das infizierte Programm und sucht sich das Code-Segment heraus, das den Virus versteckt. Dieses Segment kopiert man in die Zwischenablage mit 'Kopieren'. Dann klickt man das Diskettenfenster nach oben und eröffnet mit 'Neu' eine neue Datei. In diese neue Datei kopiert man mit 'Einfügen' das Code-Segment aus der Zwischenablage. Danach schließt man diese neue Datei und läßt die Änderungen abspeichern. Jetzt kann man das Virus-Code-Segment mit 'Löschen' endgültig aus dem infizierten Programm löschen. Auch dieses Programm schließt man und verläßt dann *ResEdit*.

Für die weitere Arbeit braucht man einen Diskettenmonitor, wie z.B. *MacTools* oder *FEdit*. Diesen startet man und öffnet die zuvor angelegte Datei mit dem Virus-Code. An der Stelle *'00A2'* in Block 1 (oder an der absoluten Stelle *02A2*) findet man dann den ursprünglichen Eintrag aus der Sprungtabelle (s. Bild 4). In diesem Fall lautete der ursprüngliche Eintrag: Springe an die Stelle *'0000'* im Segment *'0001'*. Diesen Eintrag notiert man sich und startet wieder *ResEdit*. Jetzt braucht man nur noch den Viruseintrag in der Sprungtabelle zu löschen und stattdessen die notierten Werte einzutragen. Die Datei mit dem Virus-Segment kann man danach natürlich löschen.

Zum Abschluß noch eine Warnung: Arbeiten Sie nur mit Kopien. Ich habe auf die beschriebene Weise zwar alle meine Programme gesäubert und sie funktionieren auch wieder, aber ich übernehme natürlich keine Garantie für diese Methode.

Und noch ein Tip: Kopieren Sie nach dem Start von Aladin alle Systemdateien auf die SuperDisk und booten sie dann von dieser. So kann normalerweise zumindest Ihre Systemdiskette nicht infiziert werden. Bei manchen Programmen läßt sich das Arbeiten mit dem System von der Diskette leider nicht vermeiden, wie z.B. beim Programm *'GemLoad'*. Hier wird eine leere SuperDisk vorausgesetzt...

Bernhard Malle

Literatur:

Schirmacher, Arne:

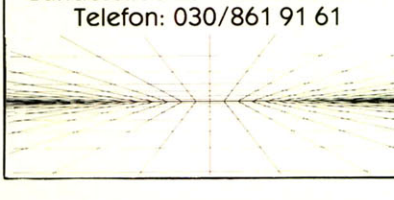
Einkaufsführer

Hier finden Sie Ihren Atari Fachhändler

1000 Berlin



DATAPLAY
Bundesallee 25 · 1000 Berlin 31
Telefon: 030/861 91 61





Keithstr. 18-20 • 1000 Berlin 30
030/21 390 21
186 346 com d

ATARI
... wir machen Spitzentechnologie preiswert.
Vertragshändler
UNION ZEISS
Kurfürstendamm 57 • 1000 Berlin 15
Telefon 32 30 61

1000 Berlin



Computer & Zubehör-Shop
Gerhard u. Bernd Waller GbR
Kieler Straße 623
2000 Hamburg 54
☎ 040/570 60 07
BTX 040 570 52 75

Ihr Atari-Händler in Berlin
COMPUTERSHOP
Radtke u. Kögel
Riesen Software Angebot
Fürbringerstr. 26 · 1000 Berlin 61
Tel. (030) 6 91 46 29 · BTX (030) 6 91 76 66

COMPUTER-STUDIO


Ihr Spezialist in Berlin für Hardware + Zubehör Eigenes Softwarestudio über 1000 verschiedene Titel am Lager

ATARI-Fachmarkt
NEC-Fachhandel · MS-DOS Fachmarkt
Katzbachstraße 6 + 8 · 1000 Berlin 61
☎ 030/786 43 40

1000 Berlin



PUBLIC DOMAIN SERVICE
Sämtliche verfügbaren ST COMPUTER - PD's. vorrätig und weitere Serien für den ST, z.B. PD-Pool und ST-Vision.
1000 Berlin 65 · Pankstr. 42
Tel.: 030/465 70 28
SERVICE STATIONEN
Auch hier alle PD's vorrätig!
1/44, Lahnstrasse 94
1/20, Schönwalder Str. 65

2000 Hamburg

Bit Computer Shop
Osterstraße 173 · 2000 Hamburg 20
Telefon: 040/49 44 00
Createam
Computer Hard & Software
Bramfelder Chaussee 300 · 2000 Hamburg 71
Telefon Sa. Nr. 040/641 50 91

RADIX Bürotechnik
Heinrich-Barth-Straße 13
2000 Hamburg 13
Telefon (0 40) 44 16 95
NEU: Software Shop

GMA mbH


Systemhändler
Wandsbeker Chaussee 58
2000 Hamburg 76

Hardware
Software
Beratung
Service


ATARI Systemfachhändler
Münsterstraße 9 · 2000 Hamburg 54
Telefon 040/56 60 1-1

2000 Norderstedt



2120 Lüneburg

Sienknecht
Bürokommunikation
Beratung - Verkauf - Werkstatt

Heiligengeiststr. 20, 2120 Lüneburg
Tel. 04131/46122, Btx 402422
Mo.-Fr. 9⁰⁰-18⁰⁰ und Sa. 9⁰⁰-13⁰⁰

2210 Itzehoe

Der Computerladen
Inhaber Ulrich Bubel Martin Kopplow

Coriansberg 2 · 2210 Itzehoe
Telefon (04821) 3390/91

2300 Kiel



Die Welt der Computer
Dreiecksplatz Nr. 7
2300 Kiel 1 · ☎ 0431/567042

2800 Bremen



Faulenstraße 48—52
2800 Bremen 1
Telefon (0421) 170577

2940 Wilhelmshaven

Radio Tiemann
ATARI-Systemfachhändler

Markstr. 52
2940 Wilhelmshaven
Telefon 04421-26145

3000 Hannover



DATALOGIC
COMPUTERSYSTEME
ATARI ST- COMPUTER HARDWARE SOFTWARE
BERATUNG SERVICE VERKAUF
CALENBERGER STR. 26
3000 HANNOVER 1
TEL.: 0511 - 32 64 89

3000 Hannover



IBM · EPSON · TRIUMPH ADLER
HEWLETT PACKARD · ATARI etc.
Computer

trendDATA Computer GmbH
Am Marstall 18–22 · 3000 Hannover 1
Telefon (0511) 16605-0

COM DATA

Am Schiffgraben 19 · 3000 Hannover 1
Telefon 0511-326736

3040 Soltau

F & T Computervertrieb

Am Hornberg 1
(Industriegeb. Almhöhe)
3040 Soltau
Tel. 05191/16522

3150 Peine

Wieckenberg & Schrage GmbH
Computertechnik
Hard- u. Software

Woltorfer Str. 8, 3150 Peine
Tel. 05171/6052/3 o. 05173/7909

3170 Gifhorn

C OMPUTER
H AUS
G IFHORN
INHABER AXEL RITZ

D-3170 GIFHORN
POMMERNRING 38
TELEFON (05371) 54498

MITGLIED DER
COMTERM
DIE COMPUTER-PARTNER
IHR FACHHÄNDLER FÜR ATARI, AMSTRAD, AEG, LEO, NEC, OKI, EPSON

Bei uns werben bringt

GEWINN

Sprechen Sie mit uns.
Heim Verlag ☎ 06151/56057

BUF

3400 Göttingen

Büroeinrichtungs-Zentrum



3400 Göttingen-Weende
Wagenstieg 14 – Tel. 0551/3857-0

3500 Kassel

Hermann Fischer GmbH
autorisierter ATARI-Fachhändler

Rudolf-Schwander-Str. 5-13
3500 Kassel
Telefon (0561) 700000

4000 Düsseldorf

HOCO
EDV ANLAGEN GMBH
Ellerstraße 155
4000 Düsseldorf 1
Telefon 0211/785213

Hard und Software
Werner Wohlfahrtstätter

Atari
Public Domain
Atari Spiele
Atari Anwender

Ladenlokal
Irenenstraße 76c
4000 Düsseldorf-Unterrath
Telefon (0211) 429876

BERNSHAUS GmbH
Bürotechnik – Bürobedarf
Cäcilienstraße 2
4000 Düsseldorf 13 (Benrath)
Telefon 0211-719181

4010 Hilden

Beachten Sie
unsere Anzeige
in diesem Heft!



Computer · Drucker · Software · Bücher · Service
Gustav-Mahler-Straße 42-44
Tel. (02103) 31880+41226

4150 KREFELD

DTP-Center
Software
Hardware
Service



C.O.P
Computer-Service GmbH
Tannenstr. 103
4150 Krefeld
☎ (02151) 77 30 42

4300 Essen

ATARI Systemfachhändler



KARSTADT Aktiengesellschaft
Limbecker Platz 4300 Essen 1
Tel.: (0201) 176399

4320 Hattingen



4330 Mülheim



Computer und Bürotechnik Vertriebsgesellschaft mbH
Dickswall 79 4330 Mülheim Telefon 02 08/3 40 34

Computer Hard- und Software auch im Leasing
Computerkurse für Anfänger und Fortgeschrittene

Tandon · VICTOR · SEL-Fernkopierer · NEC · ATARI · OKI

4422 Ahaus

ATARI · Epson · Fujitsu Molecular · NCR · Tandon · Schneider · Star

OCB

OCB-Computershop Wallstraße 3 4422 Ahaus Tel. 0 25 61/50 21	OCB-Hard- und Software Wessumerstraße 49 4422 Ahaus Tel. 0 25 61/50 21

4430 Steinfurt

CBS GmbH
COMPUTERSYSTEME
Tecklenburger Str. 27
4430 Steinfurt-Burgsteinfurt
☎ 02551/2555

4500 Osnabrück

Heinicke-Electronic
Kommenderiestr. 120 · 4500 Osnabrück
Telefon 05 41 - 8 27 99
Wir liefern Micro-Computer seit 1978

4520 Melle

CBS GmbH
COMPUTERSYSTEME
4430 Steinfurt Tel. 02551/2555
Haferstraße 25 4520 Melle
Tel.: 05422/44788

4600 Dortmund

Elektronik
Computer
Fachliteratur
ATARI-System-Fachhändler
4600 Dortmund 1, Güntherstraße 75, Tel. (02 31) 57 22 84


city-elektronik

4600 Dortmund

cc Computer Studio GmbH



Atari-Systemfachhändler
PCs von Tandy Schneider Peacock
Drucker von Star Brother NEC
Elisabethstr. 5
4600 Dortmund 1
Tel. 0231/528184 Tx 822631 cccsd Fax 0231/528131

ATARI Systemfachhändler



KARSTADT Aktiengesellschaft
Kampstraße 1 · 4600 Dortmund
Telefon (02 31) 5 43 91

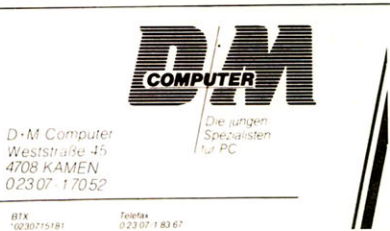
ATARI SYSTEM-Fachhändler
BÜRO STUDIO BOLZ
4600 Dortmund 1 · Brauhausstraße 4
Telefon (02 31) 5 27 713 - 16

4650 Gelsenkirchen-Horst

MENTIS GmbH
Hard- und Software, Literatur
Bauteile, Service, Versand
Groß- und Einzelhandel
Poststraße 15 · 4650 Gelsenkirchen-Horst
Telefon (02 09) 5 25 72

4650 Gelsenkirchen

C-S-A Computersysteme und Anwendung
ATARI DTP CENTER
Festplatten - Wechselplatten
SCSI - Festplatten
Scanner - Großmonitore
DTP - Vorführungen



Computersysteme und Anwendung
Hüttenstr. 56 - 4650 Gelsenkirchen
Tel.: (0209) 203420 FAX: (0209) 271584

4708 Kamen

D/M COMPUTER
Die jungen Spezialisten für PC
D·M Computer
Weststraße 45
4708 KAMEN
02307 · 17052
BTX 0230715181
Telefax 0 23 07 - 1 83 67

4712 Werne

Vogler & Trümper
Hard- und Software



Lünener Straße 14
4712 Werne
Tel. (0 23 89) 5 14 95

4800 Bielefeld

hardware software organisation service **CSF**
CSF COMPUTER & SOFTWARE GMBH
Heeper Straße 106 - 108
4800 Bielefeld 1
Tel. (05 21) 6 16 63

Carl-Severing-Str. 190
4800 Bielefeld 14
Telefon: 05 21 / 45 99 - 150
Telex: 9 37 340 krab d
Telefax: 05 21 / 45 99 - 123

MICROTEC
Software
Hardware
Beratung
Service

5000 Köln



AM RUDOLFPLATZ GmbH
5000 KÖLN 1
RICHARD-WAGNER-STR. 39
TEL. (02 21) 21 91 71

5090 Leverkusen

Rolf Rocke
Computer-Fachgeschäft
Auestraße 1
5090 Leverkusen 3
Telefon 0 21 71 / 26 24

Bei uns werben bringt
GEWINN
Sprechen Sie mit uns.
Heim Verlag 0 61 51 / 56057
BUF

5210 Troisdorf

LOGITEAM
Computerhandelsgesellschaft mbH
Kölner Straße 132
5210 Troisdorf
Tel. (0 22 41) 7 18 97
FAX (0 22 41) 7 58 58



5253 Lindlar

DIGITAL
KRÜGER
SYSTEMS
ATARI-System-Fachhändler
Hard · Software · Beratung
Komplettlösungen
5253 LINDLAR
Rheinstr 15
Tel (02266) 7894 Fax 4083

5300 Bonn

In Bonn **Ihr** Ansprechpartner für



Computer & Beratung Behnck
(0228) 67 70 21
Wir beraten und verkaufen
Mo- Fr 15-20Uhr, Sa 9-12Uhr oder nach Vereinbarung!

5414 Vallendar



GIRA**SOFT**
Systemlösungen mit **Computer**

ATARI-Systemfachhändler für Mayen-Koblenz
Ihr autorisierter Fachhändler für GTC- Personalcomputer, Star, Epson und NEC

Wir schreiben **BERATUNG** und **SERVICE** groß!
Zentrale: 5414 Vallendar, Rheinstr. 117, TEL. 0261/61727
5419 Dierdorf, Hauptstraße 50
5500 Trier, Ehrangerstr. 31

5500 Trier



Güterstraße 82 · 5500 Trier
☎ 06 51 / 20 97 10
Fordern Sie unsere Zubehör-Liste an!

5600 Wuppertal

COMPUTER FINKE



ATARI - SYSTEMFACHHÄNDLER
KIPDORF 22 · 5600 WUPPERTAL 1 · TEL. 0202 45 32 33
HARDWARE · SOFTWARE · ZUBEHÖR · SERVICE · SCHULUNGEN



... wir machen Spitzentechnologie preiswert.

MEGABYTE
Computer Vertriebs GmbH

Friedrich-Engels-Allee 162
5600 Wuppertal 2 (Barmen)
Telefon (02 02) 8 19 17

5630 Remscheid

COM SOFT
Nordstraße 57 · 5630 Remscheid
Telefon (0 21 91) 2 10 33

5650 Solingen

MegaTeam
Computer-Vertriebs-OHG
Kölbach – Finke

Hardware – Software – Zubehör – Service

Rathausstraße 1–3 · 5650 Solingen 1
Telefon (02 12) 4 58 88 · Fax (02 12) 4 73 99

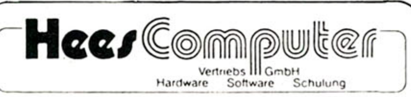
5800 Hagen



Vertragshändler Axel Böckem
Computer + Textsysteme

Eilper Str. 60 (Eilpezentrum) · 5800 Hagen
Telefon (0 23 31) 7 34 90

5900 Siegen



Siegen · Weidenauer Str. 72 · ☎ 02 71/7 34 95

6000 Frankfurt

Eickmann
Computer

***Der* Atari-System-Fachhändler !**

z.B.: Festplatten von 30 MB bis 110 MB für Atari ST und Mega ST, Zusatzaufrüstungen für Ihre Festplatten bis 110 MB, Umrüstung Ihres SM 124 in einen EM 124 Multisync für alle Auflösungen, Slotkit für PC 1...

besuchen Sie unser Fachgeschäft:
In der Römerstadt 249
6000 Frankfurt 90-Praunheim
Telefon (069) 763409

Bei uns werben bringt
GEWINN



Sprechen Sie mit uns.
Heim Verlag 0 61 51 / 56057
BUF

WAIZENEGGER
Büroeinrichtungen

Kaiserstraße 41
6000 Frankfurt/Main
Tel. (0 69) 2 73 06 - 0

DAS BÜRO-FACHGESCHÄFT!
Müller & Nemecek GmbH

Kaiserstraße 44
6000 Frankfurt/M.
Tel. (0 69) 23 25 44

6100 Darmstadt

Heim
Büro- und Computermarkt
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon (0 61 51) 5 60 57

6200 Wiesbaden



COMPUTERCENTER
DER SYNTHESIZER - STUDIO JAKOB GMBH

Computer, Software, Zubehör
MIDI - Equipment, eigener Service
6200 WIESBADEN, MAINZER STR. 137
Tel. (06121) 71 94 90 BBS: (06121) 70 17 39

6240 Königstein

KFC
COMPUTERSYSTEME
Wiesenstraße 18
6240 Königstein
Tel. 0 61 74 - 30 33
Mail-Box 0 61 74 - 53 55

6250 Limburg



6300 Gießen



6400 Fulda

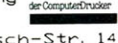
Schneider ATARI Commodore
WEINRICH
BÜRO · ORGANISATION
Ronsbachstraße 32 · 6400 Fulda
Telefon (06 61) 4 92 - 0

6457 Maintal

LANDOLT - COMPUTER



Beratung – Service
Verkauf – Leasing
Finanzierung



6457 Maintal, Robert-Bosch-Str. 14
Tel: 06181-45293 Fax: 431043

6520 Worms



Computersysteme GmbH

6520 Worms · Friedrichstraße 22

Telefon 0 62 41/67 57-58

6700 Ludwigshafen

MKV Computermarkt

Bismarck-Zentrum
6700 Ludwigshafen
Telefon 06 21 - 52 55 96

6720 Speyer

THEILLE Computersysteme
Gilgenstraße 4 · 6720 Speyer
Telefon (0 62 32) 772 16

6800 Mannheim



Computersysteme + Textsysteme
6800 Mannheim 24
Casterfeldstraße 74-76
☎ (06 21) 85 00 40 · Teletex 6 211 912



Computer-Center am Hauptbahnhof GmbH

L 14, 16-17
6800 Mannheim 1
Tel. (06 21) 2 09 83/84

6900 Heidelberg

JACOM FAMILA-CENTER
Hardware · Software
Schulung · Service

Hertzstraße 1 · 6900 Heidelberg 1
Telefon (0 62 21) 30 24 37

7000 Stuttgart



7022 L-Echterdingen

Autorisierter ATARI-System-Fachhändler
ATARI ST



Matrai Computer GmbH
Bernhauser Str 8
7022 L.-Echterdingen
☎ (07 11) 79 70 49

7030 Böblingen

Verkauf – Service – Software



COMPUTER CENTER

Norbert Hlawinka
Sindelfinger Allee 1
7030 Böblingen
Tel. 0 70 31/22 60 15

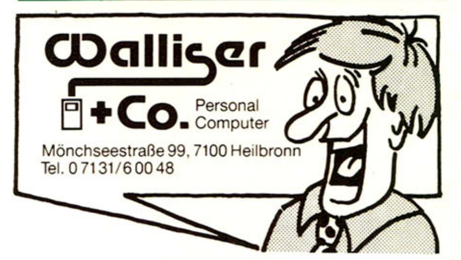
7047 Jettingen

Verkauf – Service – Software

Norbert Hlawinka
Heilbergstraße 3
Im Multi-Center
7047 Jettingen
Telefon (0 74 52) 7 76 15

mca
COMPUTER SHOP

7100 Heilbronn





Computer-Welt
Am Wollhaus 6
7100 Heilbronn
Tel. 0 71 31 - 6 84 01 - 02

7150 Backnang



7312 Kirchheim/Teck

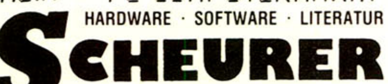


7410 Reutlingen

MKV GMBH

Listplatz 2
7410 Reutlingen
Telefon 0 71 21 - 3 66 47

Werner Brock



Untere Gerberstr. 15 · 7410 Reutlingen
Tel. 0 71 21 - 3 42 87
Tx 1 72 414 024 RMI D · box rmi taisoft · Fax 0 71 21 - 33 97 79
Autorisierter Systemfachhändler für:
ATARI, Schneider, Commodore, Panasonic, Kaypro, Sharp, NEC, OKI, STAR,...

7475 Meßstetten

Ihr ATARI-Systemhändler im Zollern-Alb-Kreis
HEIM + PC-COMPUTERMARKT
HARDWARE · SOFTWARE · LITERATUR
SCHEURER
ATARI COMMODORE CUMANA DATA-BECKER MULTITECH RITEMAN SCHNEIDER THOMSON
7475 Meßstetten 1 · Hauptstraße 10 · 0 74 31/6 12 80

7500 Karlsruhe



MKV GMBH

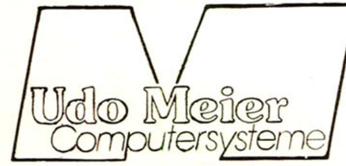
Kriegsstraße 77
7500 Karlsruhe
Telefon (07 21) 8 46 13

7600 Offenburg

FRANK LEONHARDT ELECTRONIC
Ihr Fachgeschäft für Microcomputer · Hifi · Funk

In der Jeuch 3
7600 Offenburg
Telefon 07 81/5 79 74

7700 Singen



Ringstraße 4
Telefon (0 77 31) 6 82 22

7730 VS-Schwenningen

BUS BRAUCH & SAUTER COMPUTER TECHNIK

Villinger Straße 85
7730 VS-Schwenningen
Telefon 0 77 20 / 3 80 71-72

7750 Konstanz



7800 Freiburg

Kartäuserstraße 59
7800 Freiburg
Telefon: 0761/36870-70
Fax: 0761/25849
PYRAMID COMPUTER GmbH

7850 Lörrach

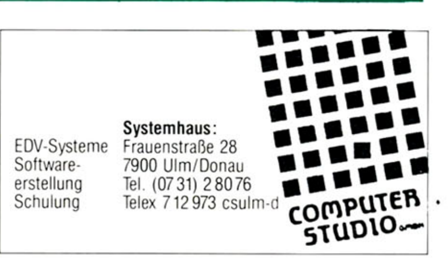


7890 Waldshut-Tiengen



service gmbh
Lenzburger Straße 4
7890 Waldshut-Tiengen
Telefon 0 77 51 / 30 94

7900 Ulm



7918 Illertissen

biotech gmbh
technische Informationsysteme
Computerladen

Marktplatz 13
7918 Illertissen
0 73 03 / 50 45

8000 München



COMPUTER + BÜROTECHNIK
COMPUTER · SOFTWARE · PERIPHERIE
BERATUNG · TECHN. KUNDENDIENST
INGOLSTÄDTER STRASSE 62L
EURO-INDUSTRIE-PARK · 8000 MÜNCHEN 45
TELEFON 089/3 11 30 66 · TELETEX 89 83 41

City Studio
Rindermarkt 6, 8000 München 2
Tel. 089 / 26 09 801 - 02

Ihr Spezialist für:
dBMAN - Komplettlösungen
vortex Massenspeicher
Vortex, dBMAN sind eingetragene Warenzeichen zugunsten Dritter
A B A C München
Kellerstraße 11, 8000 München 80
Tel. 089/ 448 99 88

Michael Weichselgartner
Ihr Spezialist für Computer
-HARDWARE -SOFTWARE
-BERATUNG -ZUBEHÖR
Chiemgaustraße 152 ▪ 8000 München 90
☎ 089 / 680 46 42

schulz computer
Schillerstraße 22
8000 München 2
Telefon (0 89) 59 73 39
Beratung · Verkauf · Kundendienst

8032 Gräfelfing



COMPUTER SYSTEME SCHULUNG
Am Haag 5
8032 Gräfelfing
Tel. 089-8 54 54 64, 85 10 43

8150 Holzkirchen



Service und Beratung sind bei uns Inklusive

8200 Rosenheim



COMPUTER + BÜROTECHNIK
Kufsteiner Str. 11, 8200 Rosenheim
Tel. (08031) 38 00 38
Fax (08031) 1 53 34

Autorisierter ATARI System-Fachhändler
Fischer & Bach
Computer GmbH
Münchner Straße 41 · 8200 Rosenheim
Tel.: 0 80 31 / 1 47 55 · Fax: 0 80 31 / 1 76 67
Hardware · Software
Schulung · Schnellreparatur

Bei uns werben bringt
GEWINN



Sprechen Sie mit uns.
Heim Verlag 0 61 51 / 56057
BUF

8400 Regensburg

Zimmermann elektroland

8400 Regensburg	8390 Passau
Dr.-Gessler-Str. 8	Kohlbruck 2a
☎ 09 41 / 9 50 85	☎ 08 51 / 5 20 07

8423 Abensberg

WITTICH COMPUTER GMBH
Tulpenstr. 16 · 8423 Abensberg
☎ 0 94 43 / 4 53



8500 Nürnberg





8520 Erlangen



Zimmermann
elektroland

8520 Erlangen
Nürnberger Straße 88
Tel. (09131) 34568

8500 Nürnberg
Hauptmarkt 17
Tel. (0911) 20798

Computerservice Decker
Meisenweg 29 - 8520 Erlangen
Telefon 09131 / 42076



Erfolgreich werben

Sprechen Sie mit uns.
Heim-Verlag ☎ (06151) 56057 BUF

8600 Bamberg



8700 Würzburg

SCHOLL
BÜROTEAM

Hardware · Software
Service · Schulung

computer center
am Dominikanerplatz
Ruf (0931) 30808-0

8720 Schweinfurt

Uhlenhuth GmbH
Computer + Unterhaltungselektronik
Albrecht-Dürer-Platz 2
8720 Schweinfurt
Telefon 09721 / 652154

8900 Augsburg



Unser Plus: Beratung u. Service
Schwalbenstr. 1 · 8900 Augsburg-Pfersee
Telefon (0821) 528533 oder 528087

Computer Vertriebs- und Software GmbH

ÖSTERREICH

A-1030 Wien

Ihr ST-Fachhändler in Wien

Computer-Studio
Wehsner Gesellschaft m.b.H

A-1030 Wien
Landstraßer Hauptstraße 2
Hilton-Einkaufspassage

A-1040 Wien

Ihr ST-Fachhändler in Wien

Computer-Studio
Wehsner Gesellschaft m.b.H

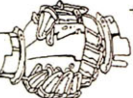
A-1040 Wien · Paniglgasse 18-20
Tel. (0222) 5057808, 5058893

A-1060 Wien



BÜROMASCHINEN IM GENERALICENTER

Ihr
ATARI Partner



Mariahilferstraße 77-79
(Generalicenter), A-1060 Wien
Tel. (0222) 961951
+
Webgasse 21, A-1060 Wien
Tel. (0222) 5976759
Generalvertretung der Bavaria-Soft

A - 1180 Wien

Ihr ST-Fachhändler in Wien



Tel. (0222) 48 52 56
A-1180 Wien - Schulgasse 63

A-8010 Graz



SCHWEIZ





Ihr Computer Spezialist

5000 Aarau, Bahnhofstrasse 86,
Tel. 064/22 78 40
4102 Basel-Binningen, Kronenplatz,
Tel. 061/47 88 64
5430 Wettingen, Zentralstrasse 93,
Tel. 056/27 16 60
8400 Winterthur, St. Gallerstrasse 41,
Tel. 052/27 96 96
8021 Zürich, Langstrasse 31,
Tel. 01/241 73 73
Grösste Auswahl an Peripherie, Software, Literatur und Zubehör.

ADAG
COMPUTER
SCHEUCHZERSTR.1 8006 ZÜRICH
TORSTRASSE25 9000 ST.GALLEN
ATARI
SOFT- UND HARDWARE
DRUCKER • ZUBEHÖR

CH-1205 Geneve

PIMENT ROUGE
INFORMATIQUE S.A.
8, RUE DES MARAICHERS
1205 GENEVE TEL. 022/28 56 24



INHALT

Ein ganz kurzes Accessory

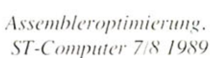
Jürgen Stessun

"Wäre doch gelacht, wenn ich da nicht noch ein paar Bytes rauskitzeln könnte", dachte ich mir, als ich mir vornahm, ein bis aufs letzte Byte optimiertes Accessorys zu schreiben, das mir den noch verbleibenden freien Speicherplatz anzeigen sollte.

Neben dem rein demonstrativen Charakter hilft Irgendwo gab's mal ein *RAM-FREE*-Programm, das auch nur den freien Restspeicher anzeigte, jedoch eine Länge von 2560 Bytes aufwies und führende Nullen nicht unterdrückte. Das geht besser, dachte ich. Mit ca. 600 Bytes war die erste Version lauffähig, aber bei weitem noch nicht fertig; nun fing die Kleinarbeit erst an:

Datenregister, wenn möglich, mit *MOVEQ*, Adreßregister mit *LEA* füllen, Unterroutinen so schieben, daß Kurzadressierung möglich wird (*BSR.S*). Quellen PC-relativ adressieren und Ziele adreßregisterindirekt ansprechen, dabei möglichst auf nicht veränderte Register zurückgreifen. Diese und einige andere auch schon in Ausgabe 7/8 1989 im Artikel Assembler-Optimierung gemachte Möglichkeiten führten dann nach langen Sitzungen zu dem Programm, was Sie unten sehen können. Hinzu kommt die Ausnutzung des Umstandes, daß nicht alle Register bei Systemaufrufen verändert werden. Aber Vorsicht, das gleicht einem Tanz auf dem Hochseil und muß bei neueren TOS-Versionen nicht so bleiben.

Interessant dürfte jedoch die Art der Initialisierung sein, wie der Aufruf des AES automatisch die drei Register *D7/A5/A6* mit Werten belegt, so daß die Parameter für CONTRL auf effektiv(st)e Weise übergeben werden können. Die Variante mit *MOVEP* (siehe 1.) zieht hierbei den Kürzeren, funktioniert leider auch nicht auf dem sonst recht guten GFA-Assembler, der hierbei falsche Adreß-Offsets berechnet. Na, und die Dezimalroutine kann sich auch sehen lassen. Sie können sie für eigene Projekte weiterverwenden.



Literatur:

Assembleroptimierung, ST-Computer 7/8 1989

M68000-Familie, W. Hilf / A. Nausch, TeWi Verlag, ISBN 3-921803-16-0

```
;******************************
; Ein ganz kurzes Accessory
; 7/89 Jürgen Stessun
; (c) MAXON Computer GmbH
; zeigt freien Speicherplatz an
;******************************
; Initialisierungsteil fürs ACC
;******************************
        lea    STACK(pc),sp ; der Stapel für's ACC
        bsr.s FILA6       ; a6 und a5 =CONTRL, d7=1
        move.w #10,(a5)+    ; 10 Opcode Appl_Init
        move.l d7,(a5)+ ; 0*Intin 1*Intout (d7=1)
        clr.w (a5)+     ; 0*Addrin
        bsr.s AES       ; AES-Aufruf, Register
                        setzen
        move.w INTOUT(pc),APPID-CONTRL(a6)
                        ; Appl.-Nummer merken
        move.w #35,(a5)+  ; 35 Menu_Register
        move.w d7,(a5)+    ; 1*Intin(d7=1)
        move.w d7,(a5)+    ; 1*Intout
        move.w d7,(a5)+    ; 1*Addrin
        move.w APPID(pc),INTIN-CONTRL(a6)
                        ; Applikationsnummer
        move.l #ACCNAME,ADDRIN-CONTRL(a6)
                        ; der ACC-Name
        bsr.s AES          ; AES aufrufen
        move.w INTOUT(pc),ACCID-CONTRL(a6)
                        ; die ACC-Nummer

LOOP:   bsr.s MEVENT      ; auf Ereignis warten
        cmpi.w #40,MSGBUFF-CONTRL(a6) ; ACC Open?
        bne.s LOOP         ; nein, weiter warten
        move.w MSGBUFF+8(pc),d0 ; die ACC-Nummer
        cmp.w ACCID(pc),d0    ; bin ich gemeint?
        bne.s LOOP   ; nein, weiter warten
        bsr.s MAIN   ; ja, Hauptroutine abarbeiten
```
→

```
         bra.s LOOP  ; und danach wieder warten

MEVENT:  move.w #23,(a5)+ ; 23 Evnt_Mesag
         move.l d7,(a5)+  ; 0*Intin 1*Intout (d7=1)
         move.w d7,(a5)+  ; 1*Addrin
         move.l #MSGBUFF,ADDRIN-CONTRL(a6)
                          ; Message-Puffer

;*********************
; AES aufrufen
;*********************
AES:     move.l #AESPB,d1   ; Zeiger auf Adressliste
         clr.w (a5)+        ; 0*Addrout
         move.w #$c8,d0     ; Wert für AES
         trap #2            ; aufrufen
FILA6:   lea   CONTRL(pc),a6  ; Zeiger für A6
         movea.l a6,a5        ; und A5
         moveq.l #1,d7      ; d7 auf 1 setzen
         rts                ; fertig mit AES
;****************************
; Der eigentliche Aktionsteil
;****************************
MAIN:    moveq.l #-1,d0     ; Restspeicher erfragen
         move.l d0,-(sp)    ; -1.L auf Stack
         move.w #$48,-(sp)  ; MALLOC
         trap #1            ; Gemdos
         addq.l #6,sp       ; in D0 Restspeicher
         lea   NUMMER(pc),a4  ; Stringadresse in
                              Alert

PRLONG:  bsr.s LONGNUM      ; in Dezimal wandeln
         move.l #ALTEXT,ADDRIN-CONTRL(a6)
                            ; Text für Alarmbox

ALERT:   move.w #52,(a5)+   ; 52 Form Alert
         move.w d7,(a5)+    ; 1*Intin(d7=1)
         move.w d7,(a5)+    ; 1*Intout
         move.w d7,(a5)+    ; 1*Addrin
         move.w d7,INTIN-CONTRL(a6) ; Default
                                      Button = 1
         bra.s AES
;**************************************
; Die Dezimal-Umwandlungsroutine
;**************************************
LONGNUM:  moveq.l #7,d4     ; insgesamt 7 Stellen
          move.l #1000,d1   ; zuerst 4-stellig
          moveq.l #(' '-'0'),d2  ; Space erlaubt
          divu.w d1,d0      ; durch 1000
          bsr.s D4TODEZ     ; umwandeln
          swap.w d0         ; Rest in d0.w
          moveq.l #100,d1   ; dann 3-stellig

D4TODEZ: move.w d0,d3       ; Zahl retten
W1:      ext.l d3           ; auf Lang erweitern
         subq.w #1,d4       ; schon letzte Ziffer?
         bne.s W2           ; nein
         clr.w d2      ; sonst kein Space erlaubt
W2:      divu.w d1,d3       ; Rest durch Teiler
         beq.s DEZPR1       ; ergab schon 0
         clr.w d2           ; kein Space erlaubt
DEZPR1:  add.b d2,d3        ; Offset zu Space
                              addieren
         addi.b #'0',d3     ; Offset für ASCII
         move.b d3,(a4)+    ; ablegen in String
         swap.w d3          ; Rest der Division
         divu.w #10,d1      ; nächster Teiler .
         bne.s W1           ; wenn ungleich 0
         rts                ; sonst fertig

         .DATA
ACCNAME: .DC.b '  Memory',0 ; 2 Spaces vor
                              'Me...'
ALTEXT:  .DC.b '[0][Memory:'
NUMMER:  .DC.b '         Bytes ' ; 8 Spaces
                                   vor 'By...'
         .DC.b '][Ok]',0
         .EVEN
AESPB:   .DC.l CONTRL,GLOBAL,INTIN,INTOUT,ADDRIN,
               ADDROUT

         .BSS
APPID:   .DS.w 1  ; die Applikationsnummer
ACCID:   .DS.w 1  ; die ACC-Nummer
MSGBUFF: .DS.b 16 ; der Nachrichtenpuffer

CONTRL:  .DS.w 1  ; Opcode der AES-Funktion
         .DS.w 1  ; Anzahl der Intin-Einträge
         .DS.w 1  ; Anzahl der Intout-Ausgaben
         .DS.w 1  ; Anzahl der Addrin-Einträge
         .DS.w 1  ; Anzahl der Addrout-Ausgaben
GLOBAL:  .DS.w 15
INTIN:   .DS.w 2  ; Wort-Eingaben
INTOUT:  .DS.w 2  ; Wort-Ausgaben
ADDRIN:  .DS.l 2  ; AdreP-Eingaben
ADDROUT: .DS.l 2  ; AdreP-Ausgaben
         .DS.b 100  ; Platz für Stapel
STACK:
         .END
```

```
a$=SPACE$(320)
adr%=VARPTR(a$)
RESTORE daten
CLR pruef%
FOR n%=0 TO 79
  READ b$
  wert%=VAL("&H"+b$)
  LPOKE adr%+4*n%,wert%
  pruef%=pruef% XOR (wert%)
NEXT n%
IF pruef%<>&H73DF00B4
  ALERT 1," Daten nicht korrekt | Bitte
  überprüfen",1,"OK",dummy%
ELSE
  BSAVE "MEMORY.ACC",adr%,318
  ALERT 1," Alles Ok ",1,"Ende",dummy%
ENDIF
EDIT
daten:
DATA 601A0000,D40000,400000,B80000,0,0,0,
4FFA01CA,61663AFC
DATA A2AC7,425D614E,3D7A0142,FFEC3AFC,
     233AC7,3AC73AC7
DATA 3D7A00F2,282D7C,D4,30612E,3D7A0122,
     FFEE6116,C6E0028
DATA FFF066F6,303A00DE,B07A00D0,66EC612A,
     60E83AFC,172AC7
DATA 3AC72D7C,118,30223C,FC,425D303C,
     C84E42,4DFA00BA,2A4E7E01
DATA 4E7570FF,2F003F3C,484E41,5C8F49FA,
     646118,2D7C0000,DD0030
DATA 3AFC0034,3AC73AC7,3AC73D47,2860BE,
     7807223C,3E8,74F080C1
DATA 61044840,72643600,48C35344,66024242,
     86C16702,4242D602
DATA 6030030,18C34843,82FC000A,66E24E75,
     20204D65,6D6F7279,5B305D
DATA 5B4D656D,6F72793A,20202020,20202020,
     42797465,73205D5B
DATA 4F6B5D00,128,132,150,154,158,160,28,
     30082A72,4040404
DATA 4000000     ! 320 Bytes=80 Data
```

Icon-Programmierung

in Pascal

Oliver Krämer

ST PASCAL PLUS ENTHÄLT BEREITS SEHR VIELE AES- UND VDI-ROUTINEN, DIE DIE PROGRAMMIERUNG EINER KOMFORTABLEN BENUTZEROBERFLÄCHE ERMÖGLICHEN. JEDOCH FINDEN SICH WEDER IM HANDBUCH DER VERSION 1.20 NOCH IM NEUEN HANDBUCH DER VERSION 2.0 HINWEISE AUF DEN UMGANG MIT ICONS.

Nach etlichen fehlgeschlagenen Versuchen (teilweise wurden sie seitens des ST mit mehr oder weniger Bomben beantwortet), diese Icons auf den Bildschirm zu bringen, stellte ich fest: und es geht doch.

Das Beispielprogramm erwartet eine Resource-Datei *ICONSMPL.RSC*, die die Icons enthalten muß. Diese werden auf dem Bildschirm dargestellt, anschließend können sie mit der Maus selektiert oder verschoben werden. Doch zuerst zum Anlegen der Resource-Datei:

Die Resource-Datei beinhaltet nur einen Dialog mit dem Namen *TDESK*, dessen Umfassung groß genug sein sollte, um zwei Icons aufnehmen zu können, und der mit dem Muster des Hintergrundes ausgefüllt sein muß (ohne Rand), damit das korrekte Löschen der Icons gewährleistet ist. In diesen Dialog plaziert man zwei Icons (arbeitet man mit dem Kuma Resource Construction Set, können als Icons z.B. das Baumicon und das Dateiicon aus dem Resourcefile des RCS verwenden, um Arbeit zu sparen) mit den Namen *ICON1* und *ICON2*. Die Namen der Objekte können natürlich auch anders gewählt werden, müssen dann allerdings auch im Programm entsprechend angepaßt werden. Bei beiden Objekte brauchen keine Objektflags gesetzt werden.

Aufbau des Programms

Die Zeilen 1-10 bedürfen wohl keiner großen Erklärung, es werden die Include-Dateien eingebunden und die Variablen deklariert. In Zeile 11 wird eine Variable *tr* vom Typ *tree* ptr deklariert, die verschiedene Informationen über einen Objektbaum und seine Kinder beinhaltet. Die für das Programm wesentlichen werden später noch erläutert.

Nach erfolgreichem Laden der Resource-Datei und nachdem die Zeigervariable *tr* durch die *find_dialog*-Routine auf den Objektbaum *tdesk* gesetzt wurde, zeichnet das Hauptprogramm zuerst durch zweimaligen Aufruf der Prozedur *Icon_Draw* die Icons mit dem Namen *Icon1* und *Icon2*. Diese Prozedur setzt zuerst die x- und y-Koordinate des "Rahmens" (Index 0) auf den Ursprung, dann die des gewählten Objekts (*item*) auf die in *xc* und *yc* übergebenen Koordinaten (weil die Koordinaten eines Objektes relativ zum Elternobjekt angegeben werden). Da *obj_draw* neben den Koordinaten auch die Breite und Höhe als Parameter verlangt, werden diese vorher von *Obj_Size* ermittelt (wobei *x* und *y* die Werte von *xc* und *yc* annehmen). Die Variablen *x,y,w,h* werden als Referenzparameter übergeben. Danach wird die Prozedur *m_contr* aufgerufen, die als Kern für die Programmierung von Icons angesehen werden kann.

Die Funktion *get_event* ermittelt Keyboard- (nur zum Verlassen der Prozedur von Bedeutung) und Mousebutton-Events. Ergibt die Überprüfung in Zeile 62 einen Mousebutton-Event, erfolgt die Bearbeitung dieses Events. Der Laufvariablen *i* wird der Index des ersten Kindes des Rahmen zugewiesen, worauf in der folgenden Repeat-Schleife der Index des Icons ermittelt wird, auf das der Mauszeiger weist. Falls nicht auf ein Icon geklickt wurde, wird die Schleife verlassen, nachdem das letzte Kind überprüft wurde (allerdings wird hier nur die oberste Kinderebene berücksichtigt, mit Hilfe der *Worktree*-Routine kann man diese Prozedur für eigene Anwendungen erweitern), wobei fl=FALSE ist. Liefert jedoch *obj_find* einen Wert, der i entspricht, und ist das gefundene Object ein Icon, wird fl=TRUE gesetzt, worauf *ic* der Wert von *i* zugewiesen wird. Eigentlich könnte man

die Schleife jetzt verlassen, jedoch sollte auch berücksichtigt werden, daß sich Icons überlappen können. Leider bietet GEM - soweit mir bekannt - keine Möglichkeit zu überprüfen, welches Icon an oberster Stelle liegt, da dies durch die verschiedene Reihenfolge der *obj_draw*-Aufrufe nicht von dem Objekt-Index der Icons abhängt. Man könnte vielleicht ein Array anlegen, in dem alle Icons mit Index aufgeführt sind und die in dieser festgelegten Reihenfolge gezeichnet werden.

Wenn nun ein Icon mit der linken Maustaste angeklickt worden ist (Überprüfung in Zeile 69), wird zunächst das selektierte Icon invers dargestellt. Dann wird mit *graf_mkstate* geprüft, ob die linke Maustaste noch gedrückt wird. Ist dies der Fall, wird die Prozedur *icon_move* aufgerufen, die das Verschieben eines Icons übernimmt: Mit der AES-Routine *Dragbox* werden die neuen Koordinaten des Icons ermittelt, das Icon wird an der ursprünglichen Position gelöscht, worauf man die übrigen neu zeichnen muß, da eventuell auch Teile eines anderen Icons gelöscht wurden, wenn es das zu verschiebende überlagert. Schließlich wird letztgenanntes an der neuen Position gezeichnet.

Wenn man viele Icons verwendet, kostet das Neuzeichnen aller Icons relativ viel Zeit, so daß sich das Programm dann vielleicht alle Icons merken sollte, die die *obj_find*-Routine liefert, um nur diese neu zu zeichnen.

Sollte ein Mausklick erfolgen, ohne daß der Mauszeiger auf einem Icon steht, werden alle Icons normal dargestellt (Zeile 111-118).

Man gelangt aus der Routine *m_contr* wieder ins Hauptprogramm, indem man die Space-Taste drückt. Dort wartet das Programm erneut auf einen Tastendruck, worauf die Icons mit der *icon_era*-Routine gelöscht werden.

Diese Prozedur zeichnet an der Position der Icons einen Teil des Elternobjekts, so daß diese "überschrieben" werden. Deshalb ist es auch von Bedeutung, daß das Füllmuster der Umrandung das gleiche ist wie das des Desktops. Ich weiß nicht, ob dieses Verfahren das optimale ist, jedoch habe ich festgestellt, daß K-Resource anscheinend nach einer ähnlichen Methode arbeitet: Ändert man im ersten Formular des K-Resource-RSC-File das Füllmuster des Rahmens (des Elternobjekt der Icons) und speichert das RSC-File, wird der Bildschirm mit diesem Muster gefüllt, nachdem man den Resource-Editor erneut gestartet hat.

Zum Schluß wünsche ich noch viel Spaß beim Probieren und mögen die Bomben an Euch vorüber gehen!

PP

```
 1:  program icon_sample; { (c) MAXON Computer GmbH }
 2:
 3:  {$I iconsmpl.i}
 4:  {$I gemconst.pas}
 5:
 6:  type
 7:    {$I gemtype.pas}
 8:
 9:  var  c              : char;
10:       init,x,y,w,h   : integer;
11:       tr             : tree_ptr;
12:
13:  {$I gemsubs}
14:
15:  {$P-}
16:
17:  procedure graf_mkstate(var mstate:integer);
18:  var  int_in   : int_in_parms;
19:       int_out  : int_out_parms;
20:       addr_in  : addr_in_parms;
21:       addr_out : addr_out_parms;
22:  begin
23:    aes_call(79,int_in,int_out,addr_in,addr_out);
24:    mstate:=int_out[3];
25:  end;
26:
27:
28:  procedure icon_draw(var pt:tree_ptr; item,xc,
                         yc:integer);
29:  var  x,y,w,h : integer;
30:  begin
31:    pt^[0].ob_x:=0;
32:    pt^[0].ob_y:=0;
33:    pt^[item].ob_x:=xc;
34:    pt^[item].ob_y:=yc;
35:    obj_size(pt,item,x,y,w,h);
36:    obj_draw(pt,item,0,0,0,0,0);
37:  end;
38:
39:
40:  procedure icon_era(var pt:tree_ptr;item:integer);
41:  var  x,y,w,h : integer;
42:            pl : tree_ptr;
43:  begin
44:    x:=0; y:=0;
45:    pt^[0].ob_x:=0; pt^[0].ob_y:=0;
46:    obj_size(pt,item,x,y,w,h);
47:    pt^[0].ob_x:=pt^[item].ob_x;
48:    pt^[0].ob_y:=pt^[item].ob_y;
49:    obj_draw(pt,0,0,0,0,0,0);
50:  end;
51:
52:
53:  procedure icon_move(ic:integer);
54:  var  xn,yn,x,y,w,h,i : integer;
55:  begin
56:    obj_size(tr,ic,x,y,w,h);
57:    dragbox(0,0,639,399,x,y,w,h,xn,yn);
58:    icon_era(tr,ic);
59:    tr^[0].ob_x:=0;
60:    tr^[0].ob_y:=0;
61:    icon_draw(tr,ic,xn,yn);
62:    i:=tr^[0].ob_head;
63:    repeat
64:      if i<>ic then
65:        begin
66:          obj_setstate(tr,i,$00,false);
67:          obj_redraw(tr,i);
68:        end;
69:      i:=tr^[i].ob_next;
70:    until i=0;  { Index der Wurzel }
71:  end;
72:
73:
74:  procedure m_contr;
75:  var  fl             : boolean;
76:       ev,mx,my,d,x,y,w,h,xn,yn,ic,i,
77:        key,bstate : integer;
78:       msg            : message_buffer;
79:  begin
80:    repeat
81:      ev:=get_event($01|$02,$01,$01,1,0,false,d,d,
                     d,d,false,d,d,d,d,
82:                  msg,key,bstate,d,mx,my,d);
83:      if ev & $02<>0 then
84:        begin
85:          i:=tr^[0].ob_head;
86:          ic:=-1;
87:          repeat
88:            fl:=(i=obj_find(tr,i,0,mx,my)) and
                    (tr^[i].ob_type=G_Icon);
89:            if fl then
90:              begin
91:                if obj_state(tr,i) & $01<>0 then
                                         fl:=not fl;
92:                  { Wenn gefundenes Objekt bereits
                       selektiert,
93:                    dann weitersuchen }
94:                ic:=i;
95:              end;
96:            i:=tr^[i].ob_next;
97:          until i=0; { i=0: Wurzelobjekt}          →
```

```
          if (ic<>-1) and (bstate & $01<>0) then
            begin
              obj_setstate(tr,ic,$01,true);
              i:=tr^[0].ob_head;
              while i<>0 do
                begin
                  if i<>ic then obj_setstate(tr,i,
                                          $00,true);
                  i:=tr^[i].ob_next;
                end;
              graf_mkstate(bstate);
              if bstate & $01 <>0 then
                                 icon_move(ic);
            end
          else
            begin
              i:=tr^[0].ob_head;
              while i<>0 do
                begin
                  obj_setstate(tr,i,$00,true);
                  i:=tr^[i].ob_next;
                end;
            end;
        end;
    until key & $FF=32;
  end;

  {$P=}

  begin
    init:=init_gem; if init<0 then halt;
    if load_resource('iconsmpl.rsc') then
      begin
        find_dialog(tdesk,tr);
        icon_draw(tr,icon2,300,200);
        icon_draw(tr,icon1,200,200);
        obj_size(tr,icon1,x,y,w,h);
        m_contr;
        read(c);
        icon_era(tr,icon2);
        icon_era(tr,icon1);
      end;
    free_resource;
    exit_gem;
  end.

```

```
(* Resource-Set indicies fuer ICONSMPL *)

CONST
  tdesk    = 0;    (* Formular/Dialog *)
  rfill    = 0;    (* BOX in Baum TDESK *)
  icon1    = 1;    (* ICON in Baum TDESK *)
  icon2    = 2;    (* ICON in Baum TDESK *)
```

Schnelle und variable Text-Routinen

M.Malich & E.Grah

Zweck dieses Programms ist es, Text in beliebiger Grösse und unter beliebigem Winkel auf dem Bildschirm darzustellen. Zusätzlich kann die Schrift noch beliebig geneigt, also kursiv dargestellt werden. Das Programm wurde vollständig in Maschinensprache geschrieben und kann von sämtlichen Hochsprachen genutzt werden.

Die Routinen sind dabei konsequent auf Geschwindigkeit getrimmt und daher sehr schnell. Das Beispielprogramm wurde in GFA-BASIC geschrieben. So ist sehr einfach ersichtlich, wie der Aufruf der Routinen zu erfolgen hat.

Die Zeichen wurden als Polygonzüge in einer 20*40-Matrix definiert. Die Tabelle ist dabei folgendermaßen aufgebaut: Zu Anfang steht der ASCII-Code des jeweiligen Zeichens, anschließend folgt dann ein Polygonzug. Beginnend mit den x,y-Koordinaten der Startposition folgt dann eine Liste von Vektoren, jeweils zuerst die x, dann y-Ordinate. Das Ende eines Polygonzugs wird mit zwei Nullbytes gekennzeichnet. Der Abschluß eines Zeichens wird nach den zwei Nullbytes für den jeweiligen Polygonzug durch $FF markiert. Der Vorteil der Zeichendefinition in Form von Polygonzügen liegt im relativ geringen Rechenaufwand, der für die Manipulation der Zeichen erforderlich ist. Man braucht sich so nicht um jeden einzelnen Punkt des Zeichens zu kümmern, sondern nur um die jeweiligen Eckpunkte. Dies ist die übliche Technik, die von Plottern benutzt wird, um Schrift darzustellen.

Sämtliche Berechnungen beziehen sich jeweils auf einen Eckpunkt eines Polygonzugs, der die Koordinaten x,y habe.

Verkleinern/Vergrößern: Die Darstellung der Zeichen in beliebiger Größe wird durch die folgenden Formeln erreicht:

Schriftgröße in x-Richtung:
X_neu:=X * X_Size
Schriftgröße in y-Richtung:
Y_neu:=Y * Y_Size

Kursivdarstellung: Um die Schrift entsprechend zu neigen, wird die jeweilige x-Ordinate des Zeichens um einen bestimmten Betrag verschoben, der von der zugehörigen y-Ordinate abhängt:

X_neu:= X + Kursiv * Y

Rotation der Zeichen: Die Rotation eines Zeichens ist schon etwas komplexer. Denn hierbei handelt es sich um eine echte Koordinatentransformation mit Hilfe einer Rotationsmatrix. Die Matrix für eine Rotation im 2dimensionalen Raum hat dabei die folgende Form :

| Cos a -Sin α | | Sin α   Cos α |

Damit erhält man die gedrehten Koordinaten durch die Multipliaktion der Matrix mit dem entsprechenden Vektor:

$$\begin{vmatrix} \text{Cos}\ \alpha & \text{-Sin}\ \alpha \\ \text{Sin}\ \alpha & \text{Cos}\ \alpha \end{vmatrix} * \begin{vmatrix} X \\ Y \end{vmatrix} = \begin{vmatrix} X_neu \\ Y_neu \end{vmatrix}$$

Komponentenweise ergibt sich damit:

X_neu:= X * Cos α - Y * Sin α
Y_neu:= X * Sin α + Y * Cos α

Die Rotation erfolgt dabei um den Ursprung des jeweiligen Zeichens. Dieser liegt in der linken unteren Ecke. Nachdem eine Ordinate entsprechend berechnet wurde, werden zum neuen x- und y-Wert noch die Bildschirmkoordinaten hinzuaddiert. Sind die Koordinaten von jeweils 2 Punkten berechnet worden, so wird eine Linie des Zeichens auf den Bildschirm gezeichnet.

Das Maschinenprogramm besteht aus 4 Teilen:

1. der Initialisierung, in der eine Tabelle berechnet wird, die einen schnellen Zugriff auf die Daten jedes einzelnen Zeichens ermöglicht,

2. einer Routine zur Einstellung der Ausgabeparameter (Zeichenfarbe, Maske für den Linienstil und Grafikmodus),

3. einer Routine zur Einstellung der Textparameter (Sinus des Drehwinkels, Cosinus des Drehwinkels, Kursiv-Parameter, Größe in x- und y-Richtung),

4. der eigentlichen Textausgabe.

Das GFA BASIC-Modul besteht aus vier Prozeduren:

1. PROCEDURE *text_install*: lädt das Assemblerprogramm und reserviert den nötigen Speicherplatz. (Man hat hier die Wahl, ob man das Assemblerprogramm von Diskette bzw. Harddisk laden will, oder ob man es aus den DATA-Zeilen schaufeln läßt.)

2.PROCEDURE *text_color (col%, mask%, mode%)*
col%: Zeichenfarbe; mask%: Linienmaske (16-Bit Wort); mode%: Grafikmodus (0, 1, 2 oder 3)

3.PROCEDURE *text_parameter(alpha,kursiv%,x_size%, y_size%)*

alpha	: Drehwinkel in Bogenmaß (0-2π)
kursiv%	: Kursivkonstante (0-300)
x_size%,y_size%	: Zeichengröße (0-600)

Bei negativer Größe werden die Buchstaben gespiegelt.

4. PROCEDURE *text_exec (a$,x_pos%,y_pos%)*
a$: auszugebender Text; x_pos%,y_pos%: Bildschirmposition in Pixel. Die Routinen *text_color* und *text_parameter* müssen nicht vor jeder Textausgabe aufgerufen werden, sondern nur, wenn man Parameter ändern möchte.

Das kurze GFA BASIC-Demoprogramm dient zum Testen der Textroutinen. Mit den Tasten + und - läßt sich der dargestellte Text rotieren. Mit den beiden Maustasten können Sie ihn vergrößern bzw. verkleinern. Der Text folgt außerdem den Mausbewegungen. Mit ein bißchen Geschick kann man das Rotieren auch flimmerfrei machen. Scheuen Sie sich nicht, mit den Textroutinen und den Parametern zu experimentieren. Durch getrenntes Vergrößern bzw. Verkleinern der x- und y-Größe lassen sich nette Effekte erzielen (Dehnen, Stauchen, Hochklappen, ...).

PP

```
;     *************************************
;     *                                   *
;     *   SCHNELLE VARIABLE TEXTAUSGABE   *
;     *                                   *
;     *               VON                 *
;     *                                   *
;     *    M.MALICH          E.GRAH       *
;     *                                   *
;     *    (c) MAXON Computer GmbH        *
;     *                                   *
;     *************************************


INIT_LINE_A    = $A000
LINE_A         = $A003
X1             = 38
PLANES         = 24
LSTLIN         = 32
;**********************************
        JMP     INIT_POINTER
        JMP     SET_PARAMETER
        JMP     SET_COLOR
        MOVE.L  4(SP),A1            ; TEXT_ADRESSE
        MOVEM   8(SP),D0/D1         ; (X,Y)-KOORD
        LEA     KURSIV,A3
        MOVE.L  18(A3),A0           ; LINE_A0 BLOCK
        MOVEQ   #0,D2
        MOVEQ   #0,D3
        MOVEQ   #0,D4
        MOVEQ   #0,D5
PLOT_CHAR:
        LEA     POINTER,A2
        MOVEQ   #0,D6
        MOVE.B  (A1)+,D6            ; NEUES ZEICHEN
        BEQ     FERTIG
        CMP.B   #32,D6              ; BLANK ??
        BEQ     ZEICHEN_FERTIG
        ADD     D6,D6               ; * 4
        ADD     D6,D6
        ADD.L   D6,A2               ; POINTER
                                      BERECHNEN
        TST.L   (A2)                ; ZEICHEN
                                      VORHANDEN ??
        BEQ     PLOT_CHAR
        MOVE.L  (A2),A2
        ADDQ.L  #1,A2
NEW_POLYGON:
        MOVE.B  (A2)+,D2            ; POLYGONSTART
        BMI     ZEICHEN_FERTIG
        MOVE.B  (A2)+,D3
POLYGON:
        MOVE.B  (A2)+,D4            ; DRAW TO (D4,D5)
        MOVE.B  (A2)+,D5
        BNE     NEXT
        TST.B   D4                  ; POLYGONENDE ??
        BEQ     NEW_POLYGON
NEXT:
        ADD.B   D2,D4
        ADD.B   D3,D5

        MOVEM.L D0-D2/D4-D5/A0-A2,-(SP)
        MOVE    D3,D6               ; FROM_Y
        MULS    (A3),D6             ; * KURSIV
        ASR     #8,D6
        ADD     D6,D2               ; + FROM_X =
                                      FROM_X
        MOVE    D5,D6               ; TO_Y
        MULS    (A3),D6             ; * KURSIV
        ASR     #8,D6
        ADD     D6,D4               ; + TO_X = TO_X

        MOVE    D2,D6               ; FROM_X
        MULS    16(A3),D6           ; * Y_COS
        MOVE    D3,D7               ; FROM_Y
        MULS    14(A3),D7           ; * Y_SIN
        SUB     D7,D6               ; -
        ASR     #8,D6
        MOVE    D6,A6               ; = NEW FROM_X

        MOVE    D2,D6               ; FROM_X
        MULS    10(A3),D6           ; * X_SIN
        MOVE    D3,D7               ; FROM_Y
        MULS    12(A3),D7           ; * X_COS
        ADD     D7,D6               ; +
        ASR     #8,D6
        MOVE    D6,A5               ; = NEW FROM_Y

        MOVE    D4,D6               ; TO_X
        MULS    16(A3),D6           ; * Y_COS
        MOVE    D5,D7               ; TO_Y
        MULS    14(A3),D7           ; * Y_SIN
        SUB     D7,D6               ; -
        ASR     #8,D6
        MOVE    D6,A4               ; = NEW TO_X

        MULS    10(A3),D4           ; TO_X * X_SIN
        MULS    12(A3),D5           ; TO_Y * X_COS
        ADD     D4,D5               ; +
        ASR     #8,D5               ; = NEW TO_Y

        MOVE    D1,D6
        ADD     D0,A6               ; + X
        ADD     D0,A4
        SUB     A5,D6               ; Y -
        SUB     D5,D1
        EXG     D1,A6
        MOVEM   D1/D6/A4/A6,X1(A0)
        DC.W    LINE_A
        MOVEM.L (SP)+,D0-D2/D4-D5/A0-A2

        MOVE.B  D4,D2
        MOVE.B  D5,D3
        BRA     POLYGON
ZEICHEN_FERTIG:
        ADD     6(A3),D0            ; + OFFSET_X
```

→

```
        SUB         8(A3),D1            ; + OFFSET_Y
        BRA         PLOT_CHAR
FERTIG:
        RTS
;************************************
INIT_POINTER:
        LEA         POINTER,A0
        LEA         ZEICHEN,A1
        CLR.L       (A0)+
        MOVEQ       #0,D0
LOOP:
        MOVEQ       #0,D1
        MOVE.B      (A1),D1
        MOVE        D1,D2
        SUB         D0,D2
        SUBQ        #2,D2
        BMI         WEG
LOOP_2:
        CLR.L       (A0)+
        DBRA        D2,LOOP_2
WEG:
        MOVE        D1,D0
        MOVE.L      A1,(A0)+
SUCHE_CHAR:
        MOVE.B      (A1)+,D1
        CMP.B       #$FF,D1
        BNE         SUCHE_CHAR
        TST.B       -2(A1)
        BNE         SUCHE_CHAR
        TST.B       -3(A1)
        BNE         SUCHE_CHAR
        CMP.B       #$FF,(A1)
        BNE         LOOP
        MOVEQ       #1,D5               ; COLOR
        MOVEQ       #-1,D6              ; MASK
        MOVEQ       #0,D7               ; GRAPHMODE
        BRA         INIT_COLOR
;************************************
SET_COLOR:
        MOVEM       4(SP),D5-D7         ; COLOR/MASK/MODE
INIT_COLOR:
        DC.W        INIT_LINE_A
        LEA         LINE_A0,A1
        MOVE.L      A0,(A1)
        MOVEQ       #-1,D0
        MOVEM       D0/D6/D7,LSTLIN(A0)
        MOVEQ       #0,D0
        MOVEQ       #0,D1
        MOVEQ       #0,D2
        MOVEQ       #0,D3
        BTST        #0,D5
        BEQ         PLANE_1
        MOVEQ       #1,D0
PLANE_1:
        BTST        #1,D5
        BEQ         PLANE_2
        MOVEQ       #1,D1
PLANE_2:
        BTST        #2,D5
        BEQ         PLANE_3
        MOVEQ       #1,D2
PLANE_3:
        BTST        #3,D5
        BEQ         PLANE_4
        MOVEQ       #1,D3
PLANE_4:
        MOVEM       D0-D3,PLANES(A0)
        RTS
;************************************
SET_PARAMETER:
        LEA         KURSIV,A0
        MOVEM       4(SP),D0-D4         ; PARAMETER_BLOCK
        MOVEM       D2-D4,(A0)
        MOVE        D0,D5               ; SIN
        MULS        D3,D5               ; * SIZE_X
        ASR.L       #8,D5
        MOVE        D1,D6               ; COS
        MULS        D3,D6               ; * SIZE_X
        ASR.L       #8,D6
        MOVE        D1,D7               ; COS
        MULS        D4,D7               ; * SIZE_Y
        ASR.L       #8,D7
        MOVE        D7,A1
        MOVE        D0,D7               ; SIN
        MULS        D4,D7               ; * SIZE_Y
        ASR.L       #8,D7

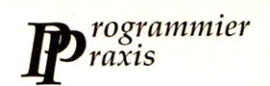
        MOVEQ       #25,D2
        EXG         D3,D4
        MULU        D2,D3
        MULS        D1,D3               ; * COS
        SWAP        D3                  ; /256/256
        MULU        D2,D4
        MULS        D0,D4               ; * SIN
        SWAP        D4                  ; /256/256
        MOVEM       D3-D7/A1,6(A0)
        RTS
;************************************

ZEICHEN:
        DC.B $21,$08,$0B,$00,$04,$04,$00,$00,$FC,$FC,
             $00,$00
        DC.B $00,$0A,$14,$FB,$14,$0B,$00,$FA,$EC,$00,
             $00,$FF
        DC.B $22,$03,$28,$05,$00,$00,$F6,$FB,$0A,$00,
             $00,$0D
        DC.B $28,$05,$00,$00,$F6,$FB,$0A,$00,$00,$FF,
             $24,$03
        DC.B $0F,$04,$FC,$07,$00,$04,$04,$00,$07,$FC,
             $04,$F9
        DC.B $00,$FC,$04,$00,$06,$04,$04,$07,$00,$04,
             $FC,$00
        DC.B $00,$0B,$28,$00,$E3,$00,$00,$FF,$25,$03,
             $0B,$0F
        DC.B $1D,$00,$00,$03,$28,$07,$00,$00,$F9,$F9,
             $00,$00
        DC.B $07,$00,$00,$0B,$12,$07,$00,$00,$F9,$F9,
             $00,$00
        DC.B $07,$00,$00,$FF,$26,$12,$16,$FC,$00,$00,
             $F9,$FC
        DC.B $FC,$FD,$00,$FC,$04,$00,$07,$0B,$08,$00,
             $06,$FC
        DC.B $04,$FB,$FC,$00,$FA,$0D,$ED,$00,$00,$FF,
             $27,$08
        DC.B $28,$05,$00,$00,$F6,$FB,$0A,$00,$00,$FF,
             $28,$06
        DC.B $0B,$FD,$09,$00,$0B,$03,$09,$00,$00,$FF,
             $29,$0F
        DC.B $0B,$03,$09,$00,$0B,$FD,$09,$00,$00,$FF,
             $2A,$03
        DC.B $1A,$10,$00,$00,$00,$0B,$12,$00,$10,$00,
             $00,$03
        DC.B $12,$10,$10,$00,$00,$03,$22,$10,$F0,$00,
             $00,$FF
        DC.B $2B,$03,$1A,$10,$00,$00,$00,$0B,$12,$00,
             $10,$00
        DC.B $00,$FF,$2C,$08,$0F,$00,$05,$04,$00,$00,
             $FB,$FD
        DC.B $FC,$01,$04,$FE,$00,$00,$00,$FF,$2D,$03,
             $1A,$0F
        DC.B $00,$00,$00,$FF,$2E,$08,$0B,$00,$04,$04,
             $00,$00
        DC.B $FC,$FC,$00,$00,$00,$FF,$2F,$03,$0B,$0F,
             $1D,$00
        DC.B $00,$FF,$30,$03,$0F,$00,$15,$04,$04,$07,
             $00,$04
        DC.B $FC,$00,$EB,$FC,$FC,$F9,$00,$FC,$04,$0F,
             $15,$00
        DC.B $00,$FF,$31,$03,$1A,$08,$0E,$00,$E3,$00,
             $00,$FF
        DC.B $32,$03,$1E,$00,$06,$04,$04,$07,$00,$04,
             $FC,$00
        DC.B $FA,$F1,$ED,$0F,$00,$00,$00,$FF,$33,$03,
             $28,$0F
        DC.B $00,$F9,$F2,$03,$00,$04,$FC,$00,$F9,$FC,
             $FC,$F9
        DC.B $00,$FC,$04,$00,$00,$FF,$34,$03,$0F,$0B,
             $19,$00
        DC.B $00,$0E,$14,$00,$F7,$00,$00,$03,$0F,$0F,
             $00,$00
        DC.B $00,$FF,$35,$03,$0F,$04,$FC,$07,$00,$04,
             $04,$00
        DC.B $07,$FC,$04,$F5,$00,$00,$0E,$0F,$00,$00,
             $00,$FF
        DC.B $36,$03,$16,$04,$04,$07,$00,$04,$FC,$00,
             $F9,$FC
        DC.B $FC,$F9,$00,$FC,$04,$00,$15,$04,$04,$07,
             $00,$04
        DC.B $FC,$00,$00,$FF,$37,$03,$28,$0F,$00,$F5,
             $E3,$00
        DC.B $00,$FF,$38,$03,$0F,$00,$07,$04,$04,$07,
             $00,$04
```

→

```
        DC.B $04,$00,$06,$FC,$04,$F9,$00,$FC,$FC,$00,$FA,$04
        DC.B $FC,$00,$00,$0E,$1A,$04,$FC,$00,$F9,$FC,$FC,$F9
        DC.B $00,$FC,$04,$00,$00,$FF,$39,$03,$0F,$04,$FC,$07
        DC.B $00,$04,$04,$00,$15,$FC,$04,$F9,$00,$FC,$FC,$00
        DC.B $FA,$04,$FC,$0B,$00,$00,$00,$FF,$3A,$08,$19,$00
        DC.B $05,$04,$00,$00,$FB,$FC,$00,$00,$00,$08,$0F,$00
        DC.B $05,$04,$00,$00,$FB,$FC,$00,$00,$00,$FF,$3B,$08
        DC.B $19,$00,$05,$04,$00,$00,$FB,$FC,$00,$00,$00,$08
        DC.B $0F,$00,$05,$04,$00,$00,$FB,$FD,$FC,$01,$04,$FE
        DC.B $00,$00,$00,$FF,$3C,$0A,$10,$F9,$07,$07,$07,$00
        DC.B $00,$FF,$3D,$03,$1D,$0F,$00,$00,$00,$12,$17,$F1
        DC.B $00,$00,$00,$FF,$3E,$0B,$1E,$07,$F9,$F9,$F9,$00
        DC.B $00,$FF,$3F,$08,$0B,$00,$04,$04,$00,$00,$FC,$FC
        DC.B $00,$00,$00,$08,$14,$0A,$0A,$00,$06,$FC,$04,$F9
        DC.B $00,$FC,$FC,$07,$00,$03,$FD,$FB,$F3,$00,$00,$FF
        DC.B $41,$03,$0B,$00,$19,$04,$04,$07,$00,$04,$FC,$00
        DC.B $E7,$00,$00,$12,$1A,$F1,$00,$00,$00,$FF,$42,$03
        DC.B $0B,$00,$1D,$0B,$00,$04,$FC,$00,$FA,$FC,$FC,$F5
        DC.B $00,$00,$00,$0E,$1A,$04,$FC,$00,$F9,$FC,$FC,$F5
        DC.B $00,$00,$00,$FF,$43,$12,$0F,$FC,$FC,$F9,$00,$FC
        DC.B $04,$00,$15,$04,$04,$07,$00,$04,$FC,$00,$00,$FF
        DC.B $44,$03,$0B,$00,$1D,$0B,$00,$04,$FC,$00,$EB,$FC
        DC.B $FC,$F5,$00,$00,$00,$FF,$45,$03,$0B,$00,$1D,$0F
        DC.B $00,$00,$00,$0E,$1A,$F5,$00,$00,$00,$03,$0B,$0F
        DC.B $00,$00,$00,$FF,$46,$03,$0B,$00,$1D,$0F,$00,$00
        DC.B $00,$0E,$1A,$F5,$00,$00,$00,$FF,$47,$0B,$1A,$07
        DC.B $00,$00,$F5,$FC,$FC,$F9,$00,$FC,$04,$00,$15,$04
        DC.B $04,$07,$00,$04,$FC,$00,$00,$FF,$48,$03,$0B,$00
        DC.B $1D,$00,$00,$12,$28,$00,$E3,$00,$00,$12,$1A,$F1
        DC.B $00,$00,$00,$FF,$49,$07,$0B,$07,$00,$00,$00,$0B
        DC.B $0B,$00,$1D,$00,$00,$07,$28,$07,$00,$00,$00,$FF
        DC.B $4A,$03,$0F,$04,$FC,$07,$00,$04,$04,$00,$19,$00
        DC.B $00,$FF,$4B,$03,$0B,$00,$1D,$00,$00,$12,$28,$F9
        DC.B $F2,$F8,$00,$00,$00,$0B,$1A,$07,$F1,$00,$00,$FF
        DC.B $4C,$03,$28,$00,$E3,$0F,$00,$00,$00,$FF,$4D,$03
        DC.B $0B,$00,$1D,$08,$F2,$07,$0E,$00,$E3,$00,$00,$FF
        DC.B $4E,$03,$0B,$00,$1D,$0F,$E3,$00,$1D,$00,$00,$FF
        DC.B $4F,$03,$0F,$00,$15,$04,$04,$07,$00,$04,$FC,$00
        DC.B $EB,$FC,$FC,$F9,$00,$FC,$04,$00,$00,$FF,$50,$03
        DC.B $0B,$00,$1D,$0B,$00,$04,$FC,$00,$FA,$FC,$FC,$F5
        DC.B $00,$00,$00,$FF,$51,$03,$0F,$00,$15,$04,$04,$07
        DC.B $00,$04,$FC,$00,$EB,$FC,$FC,$F9,$00,$FC,$04,$00
        DC.B $00,$0E,$0F,$04,$FC,$00,$00,$FF,$52,$03,$0B,$00
        DC.B $1D,$0B,$00,$04,$FC,$00,$FA,$FC,$FC,$F5,$00,$00
        DC.B $00,$0E,$1A,$04,$F1,$00,$00,$FF,$53,$03,$0F,$04
        DC.B $FC,$07,$00,$04,$04,$00,$07,$FC,$04,$F9,$00,$FC
        DC.B $04,$00,$06,$04,$04,$07,$00,$04,$FC,$00,$00,$FF
        DC.B $54,$0B,$0B,$00,$1D,$00,$00,$03,$28,$0F,$00,$00
        DC.B $00,$FF,$55,$03,$28,$00,$E7,$04,$FC,$07,$00,$04
        DC.B $04,$00,$19,$00,$00,$FF,$56,$03,$28,$08,$E3,$07
        DC.B $1D,$00,$00,$FF,$57,$03,$28,$00,$E7,$04,$FC,$04
        DC.B $04,$03,$FC,$04,$04,$00,$19,$00,$00,$FF,$58,$03
        DC.B $28,$0F,$E3,$00,$00,$03,$0B,$0F,$1D,$00,$00,$FF
        DC.B $59,$03,$28,$08,$F2,$07,$0E,$00,$00,$0B,$1A,$00
        DC.B $F1,$00,$00,$FF,$5A,$03,$28,$0F,$00,$F1,$E3,$0F
        DC.B $00,$00,$00,$FF,$61,$03,$0F,$00,$07,$04,$04,$07
        DC.B $00,$04,$FC,$00,$00,$12,$1A,$00,$F1,$00,$00,$12
        DC.B $0F,$FC,$FC,$F9,$00,$FC,$04,$00,$00,$FF,$62,$03
        DC.B $0B,$00,$1D,$00,$00,$03,$16,$04,$04,$07,$00,$04
        DC.B $FC,$00,$F9,$FC,$FC,$F9,$00,$FC,$04,$00,$00,$FF
        DC.B $63,$12,$0F,$FC,$FC,$F9,$00,$FC,$04,$00,$07,$04
        DC.B $04,$07,$00,$04,$FC,$00,$00,$FF,$64,$12,$0F,$FC
        DC.B $FC,$F9,$00,$FC,$04,$00,$07,$04,$04,$07,$00,$04
        DC.B $FC,$00,$00,$12,$28,$00,$E3,$00,$00,$FF,$65,$12
        DC.B $0F,$FC,$FC,$F9,$00,$FC,$04,$00,$07,$04,$04,$07
        DC.B $00,$04,$FC,$00,$FC,$F1,$00,$00,$00,$FF,$66,$07
        DC.B $0B,$00,$19,$04,$04,$03,$00,$04,$FC,$00,$00,$0B
        DC.B $1A,$F8,$00,$00,$00,$FF,$67,$03,$05,$04,$FC,$07
        DC.B $00,$04,$04,$00,$15,$00,$00,$12,$16,$FC,$04,$F9
        DC.B $00,$FC,$FC,$00,$F9,$04,$FC,$07,$00,$04,$04,$00
        DC.B $00,$FF,$68,$03,$0B,$00,$1D,$00,$00,$03,$16,$04
        DC.B $04,$07,$00,$04,$FC,$00,$F5,$00,$00,$FF,$69,$07
        DC.B $0B,$07,$00,$00,$00,$0B,$0B,$00,$0F,$FC,$00,$00
        DC.B $00,$0B,$1F,$00,$03,$00,$00,$FF,$6A,$05,$05,$04
        DC.B $FC,$03,$00,$04,$04,$00,$15,$00,$00,$10,$1F,$00
        DC.B $03,$00,$00,$FF,$6B,$03,$0B,$00,$1D,$00,$00,$12
        DC.B $1B,$F1,$F8,$0F,$F8,$00,$00,$FF,$6C,$07,$0B,$07
        DC.B $00,$00,$00,$0B,$0B,$00,$1D,$FC,$00,$00,$00,$FF
        DC.B $6D,$03,$0B,$00,$0F,$00,$00,$03,$16,$04,$04,$04
        DC.B $FC,$00,$F5,$00,$00,$0B,$16,$04,$04,$04,$FC,$00
        DC.B $F5,$00,$00,$FF,$6E,$03,$0B,$00,$0F,$00,$00,$03
        DC.B $16,$04,$04,$07,$00,$04,$FC,$00,$F5,$00,$00,$FF
        DC.B $6F,$03,$0F,$00,$07,$04,$04,$07,$00,$04,$FC,$00
        DC.B $F9,$FC,$FC,$F9,$00,$FC,$04,$00,$00,$FF,$70,$03
        DC.B $01,$00,$19,$00,$00,$03,$16,$04,$04,$07,$00,$04
        DC.B $FC,$00,$F9,$FC,$FC,$F9,$00,$FC,$04,$00,$00,$FF
```

→




```
        DC.B$71,$12,$0F,$FC,$FC,$F9,$00,$FC,$04,$00,$07,$04
        DC.B$04,$07,$00,$04,$FC,$00,$00,$12,$1A,$00,$E7,$00
        DC.B$00,$FF,$72,$03,$0B,$00,$0F,$00,$00,$03,$16,$04
        DC.B $04,$07,$00,$04,$FC,$00,$00,$FF,$73,$03,$0F,$04
        DC.B $FC,$07,$00,$04,$04,$FC,$04,$F8,$00,$FD,$03,$04
        DC.B $04,$07,$00,$04,$FC,$00,$00,$FF,$74,$12,$0F,$FC
        DC.B $FC,$FC,$00,$FC,$04,$00,$19,$00,$00,$03,$1A,$06
        DC.B $00,$00,$00,$FF,$75,$03,$1A,$00,$F5,$04,$FC,$07
        DC.B $00,$04,$04,$00,$00,$12,$0B,$00,$0F,$00,$00,$FF
        DC.B $76,$03,$1A,$07,$F1,$01,$00,$07,$0F,$00,$00,$FF
        DC.B $77,$03,$1A,$00,$F5,$04,$FC,$04,$04,$04,$FC,$04
        DC.B $04,$00,$0B,$00,$00,$FF,$78,$03,$1A,$0F,$F1,$00
        DC.B $00,$03,$0B,$0F,$0F,$00,$00,$FF,$79,$03,$05,$04
        DC.B $FC,$07,$00,$04,$04,$00,$15,$00,$00,$03,$1A,$00
        DC.B $F5,$04,$FC,$07,$00,$04,$04,$00,$00,$FF,$7A,$03
        DC.B $1A,$0F,$00,$F1,$F1,$0F,$00,$00,$00,$FF,$81,$03
        DC.B $1A,$00,$F5,$04,$FC,$07,$00,$04,$04,$00,$00,$12
        DC.B $0B,$00,$0F,$00,$00,$07,$1C,$00,$03,$00,$00,$0E
        DC.B $1C,$00,$03,$00,$00,$FF,$84,$03,$0F,$00,$07,$04
        DC.B $04,$07,$00,$04,$FC,$00,$00,$12,$1A,$00,$F1,$00
        DC.B $00,$12,$0F,$FC,$FC,$F9,$00,$FC,$04,$00,$00,$07
        DC.B $1C,$00,$03,$00,$00,$0E,$1C,$00,$03,$00,$00,$FF
        DC.B $8E,$03,$0B,$00,$19,$04,$04,$07,$00,$04,$FC,$00
        DC.B $E7,$00,$00,$12,$1A,$F1,$00,$00,$00,$07,$2B,$00
        DC.B $03,$00,$00,$0E,$2B,$00,$03,$00,$00,$FF,$94,$03
        DC.B $0F,$00,$07,$04,$04,$07,$00,$04,$FC,$00,$F9,$FC
        DC.B $FC,$F9,$00,$FC,$04,$00,$00,$07,$1C,$00,$03,$00
        DC.B $00,$0E,$1C,$00,$03,$00,$00,$FF,$99,$03,$0F,$00
        DC.B $15,$04,$04,$07,$00,$04,$FC,$00,$EB,$FC,$FC,$F9
        DC.B $00,$FC,$04,$00,$00,$07,$2B,$00,$03,$00,$00,$0E
        DC.B $2B,$00,$03,$00,$00,$FF,$9A,$03,$28,$00,$E7,$04
        DC.B $FC,$07,$00,$04,$04,$00,$19,$00,$00,$07,$2B,$00
        DC.B $03,$00,$00,$0E,$2B,$00,$03,$00,$00,$FF,$9E,$03
        DC.B $01,$00,$23,$04,$04,$07,$00,$04,$FC,$00,$FA,$FC
        DC.B $FC,$FD,$00,$00,$00,$0E,$1A,$04,$FC,$00,$F9,$FC
        DC.B $FC,$F9,$00,$00,$00,$FF,$FF

KURSIV:         DC.W        0
SIZE_X:         DC.W        0
SIZE_Y:         DC.W        0
OFFSET_X:       DC.W        0
OFFSET_Y:       DC.W        0
X_SIN:          DC.W        0
X_COS:          DC.W        0
Y_SIN:          DC.W        0
Y_COS:          DC.W        0
LINE_A0:        DC.L        0
POINTER:        DS.L        256,0

 END
```

```
' ************************************
' *                                  *
' *   SCHNELLE VARIABLE TEXTAUSGABE  *
' *                                  *
' *              VON                 *
' *                                  *
' *    M.MALICH         E.GRAH       *
' *                                  *
' *    (c) MAXON Computer GmbH       *
' *                                  *
' ************************************
'
Procedure Text_install
  Local Z%,Wert%
  Dim Text!(3500*8)
  Text_init%=Varptr(Text!(0))
  Text_parameter%=Text_init%+4
  Text_color%=Text_init%+8
  Text_exec%=Text_init%+12
  ' *********************************************
  ' **    Entweder die Maschinenspracheroutinen
  ' **    werden von Diskette bzw. HD geladen,
  ' *********************************************
  '  Bload "TEXT.B",Text_init%
  ' *********************************************
  ' **   oder der Maschinencode wird aus den
  ' **   DATA-Zeilen eingelesen.
  ' *********************************************
  Restore Text_daten
  T%=Text_init%
  Do
    Read Wert%
    Exit If Wert%=-1
    Poke T%,Wert%
    Inc T%
  Loop
  '
  '
  Void C:Text_init%()
Return
'
Procedure Text_color(Col%,Mask%,Mode%)
  Void C:Text_color%(Col%,Mask%,Mode%)
Return
'
Procedure Text_parameter(Alpha,Kursiv%,X_size%,Y_size%)
  Void C:Text_parameter%(Sin(Alpha)*256,Cos(Alpha)*256,Kursiv%,X_size%,Y_size%)
Return
'
Procedure Text_exec(A$,X_pos%,Y_pos%)
  Void C:Text_exec%(L:Varptr(A$),X_pos%,Y_pos%)
Return
'
'
' *********************************************
' **   Die folgenden DATA-Zeilen werden nur
' **   benötigt, wenn man die Maschinensprache-
' **   routinen nicht von Diskette bzw. HD lädt.
' *********************************************
'
Text_daten:
'
Data 78,250,0,226,78,250,1,116,78,250,1,34,34,111,0,4,76
Data 175,0,3,0,8,71,250,9,6,32,107,0,18,116,0,118,0,120
Data 0,122,0,69,250,9,12,124,0,28,25,103,0,0,178,188,60
Data 0,32,103,0,0,158,220,70,220,70,213,198,74,146,103
Data 226,36,82,82,138,20,26,107,0,0,138,22,26,24,26,26
Data 26,102,0,0,6,74,4,103,236,216,2,218,3,72,231,236,224
Data 60,3,205,211,224,70,212,70,60,5,205,211,224,70,216
Data 70,60,2,205,235,0,16,62,3,207,235,0,14,156,71,224
Data 70,60,70,60,2,205,235,0,10,62,3,207,235,0,12,220,71
Data 224,70,58,70,60,4,205,235,0,16,62,5,207,235,0,14,156
Data 71,224,70,56,70,201,235,0,10,203,235,0,12,218,68,224
```

→

```
74: Data 69,60,1,220,192,216,192,156,77,146,69,195,
         142,72,168
75: Data 80,66,0,38,160,3,76,223,7,55,20,4,22,5,96,0,
         255,124
76: Data 208,107,0,6,146,107,0,8,96,0,255,70,78,117,
         65,250
77: Data 8,78,67,250,0,208,66,152,112,0,114,0,18,17,
         52,1,148
78: Data 64,85,66,107,0,0,8,66,152,81,202,255,252,48,
         1,32,201
79: Data 18,25,178,60,0,255,102,248,74,41,255,254,102,
         242,74
80: Data 41,255,253,102,236,12,17,0,255,102,206,122,1,
         124,255
81: Data 126,0,96,0,0,8,76,175,0,224,0,4,160,0,67,250,
         7,250
82: Data 34,136,112,255,72,168,0,193,0,32,112,0,114,0,
         116,0
83: Data 118,0,8,5,0,0,103,0,0,4,112,1,8,5,0,1,103,0,
         0,4,114
84: Data 1,8,5,0,2,103,0,0,4,116,1,8,5,0,3,103,0,0,4,
         118,1
85: Data 72,168,0,15,0,24,78,117,65,250,7,162,76,175,
         0,31,0
86: Data 4,72,144,0,28,58,0,203,195,224,133,60,1,205,
         195,224
87: Data 134,62,1,207,196,224,135,50,71,62,0,207,196,
         224,135
88: Data 116,25,201,67,198,194,199,193,72,67,200,194,
         201,192
89: Data 72,68,72,168,2,248,0,6,78,117,33,8,11,0,4,4,
         0,0,252
90: Data 252,0,0,0,10,20,251,20,11,0,250,236,0,0,255,
         34,3,40
91: Data 5,0,0,246,251,10,0,0,13,40,5,0,0,246,251,10,
         0,0,255
92: Data 36,3,15,4,252,7,0,4,4,0,7,252,4,249,0,252,4,
         0,6,4
93: Data 4,7,0,4,252,0,0,11,40,0,227,0,0,255,37,3,11,
         15,29
94: Data 0,0,3,40,7,0,0,249,249,0,0,7,0,0,11,18,7,0,0,
         249,249
95: Data 0,0,7,0,0,255,38,18,22,252,0,0,249,252,252,
         253,0,252
96: Data 4,0,7,11,8,0,6,252,4,251,252,0,250,13,237,0,
         0,255
97: Data 39,8,40,5,0,0,246,251,10,0,0,255,40,6,11,253,
         9,0,11
98: Data 3,9,0,0,255,41,15,11,3,9,0,11,253,9,0,0,255,
         42,3,26
99: Data 16,0,0,0,11,18,0,16,0,0,3,18,16,16,0,0,3,34,
         16,240
100: Data 0,0,255,43,3,26,16,0,0,0,11,18,0,16,0,0,255,
         44,8,15
101: Data 0,5,4,0,0,251,253,252,1,4,254,0,0,0,255,45,3,
         26,15
102: Data 0,0,0,255,46,8,11,0,4,4,0,0,252,252,0,0,0,
         255,47,3
103: Data 11,15,29,0,0,255,48,3,15,0,21,4,4,7,0,4,252,
         0,235
104: Data 252,252,249,0,252,4,15,21,0,0,255,49,3,26,8,
         14,0,227
105: Data 0,0,255,50,3,30,0,6,4,4,7,0,4,252,0,250,241,
         237,15
106: Data 0,0,0,255,51,3,40,15,0,249,242,3,0,4,252,0,
         249,252
107: Data 252,249,0,252,4,0,0,255,52,3,15,11,25,0,0,14,
         20,0
108: Data 247,0,0,3,15,15,0,0,0,255,53,3,15,4,252,7,0,
         4,4,0
109: Data 7,252,4,245,0,0,14,15,0,0,0,255,54,3,22,4,4,
         7,0,4
110: Data 252,0,249,252,252,249,0,252,4,0,21,4,4,7,0,4,
         252,0
111: Data 0,255,55,3,40,15,0,245,227,0,0,255,56,3,15,0,
         7,4,4
112: Data 7,0,4,4,0,6,252,4,249,0,252,252,0,250,4,252,
         0,0,14
113: Data 26,4,252,0,249,252,252,249,0,252,4,0,0,255,
         57,3,15
114: Data 4,252,7,0,4,4,0,21,252,4,249,0,252,252,0,250,
         4,252
115: Data 11,0,0,0,255,58,8,25,0,5,4,0,0,251,252,0,0,0,
         8,15
116: Data 0,5,4,0,0,251,252,0,0,0,255,59,8,25,0,5,4,0,
         0,251
117: Data 252,0,0,0,8,15,0,5,4,0,0,251,253,252,1,4,254,
         0,0,0
118: Data 255,60,10,16,249,7,7,7,0,0,255,61,3,29,15,0,
         0,0,18
119: Data 23,241,0,0,0,255,62,11,30,7,249,249,249,0,0,
         255,63
120: Data 8,11,0,4,4,0,0,252,252,0,0,0,8,20,10,10,0,6,
         252,4
121: Data 249,0,252,252,7,0,3,253,251,243,0,0,255,65,3,
         11,0
122: Data 25,4,4,7,0,4,252,0,231,0,0,18,26,241,0,0,0,
         255,66
123: Data 3,11,0,29,11,0,4,252,0,250,252,252,245,0,0,0,
         14,26
124: Data 4,252,0,249,252,252,245,0,0,0,255,67,18,15,
         252,252
125: Data 249,0,252,4,0,21,4,4,7,0,4,252,0,0,255,68,3,
         11,0,29
126: Data 11,0,4,252,0,235,252,252,245,0,0,0,255,69,3,
         11,0,29
127: Data 15,0,0,0,14,26,245,0,0,0,3,11,15,0,0,0,255,
         70,3,11
128: Data 0,29,15,0,0,0,14,26,245,0,0,0,255,71,11,26,7,
         0,0,245
129: Data 252,252,249,0,252,4,0,21,4,4,7,0,4,252,0,0,
         255,72
130: Data 3,11,0,29,0,0,18,40,0,227,0,0,18,26,241,0,0,
         0,255
131: Data 73,7,11,7,0,0,0,11,11,0,29,0,0,7,40,7,0,0,0,
         255,74
132: Data 3,15,4,252,7,0,4,4,0,25,0,0,255,75,3,11,0,29,
         0,0,18
133: Data 40,249,242,248,0,0,0,11,26,7,241,0,0,255,76,
         3,40,0
134: Data 227,15,0,0,0,255,77,3,11,0,29,8,242,7,14,0,
         227,0,0
135: Data 255,78,3,11,0,29,15,227,0,29,0,0,255,79,3,15,
         0,21
136: Data 4,4,7,0,4,252,0,235,252,252,249,0,252,4,0,0,
         255,80
137: Data 3,11,0,29,11,0,4,252,0,250,252,252,245,0,0,0,
         255,81
138: Data 3,15,0,21,4,4,7,0,4,252,0,235,252,252,249,0,
         252,4
139: Data 0,0,14,15,4,252,0,0,255,82,3,11,0,29,11,0,4,
         252,0
140: Data 250,252,252,245,0,0,0,14,26,4,241,0,0,255,83,
         3,15
141: Data 4,252,7,0,4,4,0,7,252,4,249,0,252,4,0,6,4,4,
         7,0,4
142: Data 252,0,0,255,84,11,11,0,29,0,0,3,40,15,0,0,0,
         255,85
143: Data 3,40,0,231,4,252,7,0,4,4,0,25,0,0,255,86,3,
         40,8,227
144: Data 7,29,0,0,255,87,3,40,0,231,4,252,4,4,3,252,4,
         4,0,25
145: Data 0,0,255,88,3,40,15,227,0,0,3,11,15,29,0,0,
         255,89,3
146: Data 40,8,242,7,14,0,0,11,26,0,241,0,0,255,90,3,
         40,15,0
147: Data 241,227,15,0,0,0,255,97,3,15,0,7,4,4,7,0,4,
         252,0,0
148: Data 18,26,0,241,0,0,18,15,252,252,249,0,252,4,0,
         0,255
149: Data 98,3,11,0,29,0,0,3,22,4,4,7,0,4,252,0,249,
         252,252
150: Data 249,0,252,4,0,0,255,99,18,15,252,252,249,0,
         252,4,0
151: Data 7,4,4,7,0,4,252,0,0,255,100,18,15,252,252,
         249,0,252
152: Data 4,0,7,4,4,7,0,4,252,0,0,18,40,0,227,0,0,255,
         101,18
153: Data 15,252,252,249,0,252,4,0,7,4,4,7,0,4,252,0,
         252,241
154: Data 0,0,0,255,102,7,11,0,25,4,4,3,0,4,252,0,0,11,
         26,248
155: Data 0,0,0,255,103,3,5,4,252,7,0,4,4,0,21,0,0,18,
         22,252
156: Data 4,249,0,252,252,0,249,4,252,7,0,4,4,0,0,255,
         104,3
157: Data 11,0,29,0,0,3,22,4,4,7,0,4,252,0,245,0,0,255,
         105,7
158: Data 11,7,0,0,0,11,11,0,15,252,0,0,0,11,31,0,3,0,
         0,255
159: Data 106,5,5,4,252,3,0,4,4,0,21,0,0,16,31,0,3,0,0,
         255,107
```

→

```
Data 3,11,0,29,0,0,18,27,241,248,15,248,0,0,255,
     108,7,11
Data 7,0,0,0,11,11,0,29,252,0,0,0,255,109,3,11,0,
     15,0,0
Data 3,22,4,4,4,252,0,245,0,0,11,22,4,4,4,252,0,
     245,0,0
Data 255,110,3,11,0,15,0,0,3,22,4,4,7,0,4,252,0,
     245,0,0
Data 255,111,3,15,0,7,4,4,7,0,4,252,0,249,252,252,
     249,0
Data 252,4,0,0,255,112,3,1,0,25,0,0,3,22,4,4,7,0,
     4,252
Data 0,249,252,252,249,0,252,4,0,0,255,113,18,15,
     252,252
Data 249,0,252,4,0,7,4,4,7,0,4,252,0,0,18,26,0,
     231,0,0
Data 255,114,3,11,0,15,0,0,3,22,4,4,7,0,4,252,0,0,
     255,115
Data 3,15,4,252,7,0,4,4,252,4,248,0,253,3,4,4,7,0,
     4,252
Data 0,0,255,116,18,15,252,252,252,0,252,4,0,25,0,
     0,3,26
Data 6,0,0,0,255,117,3,26,0,245,4,252,7,0,4,4,0,0,
     18,11
Data 0,15,0,0,255,118,3,26,7,241,1,0,7,15,0,0,255,
     119,3
Data 26,0,245,4,252,4,4,4,252,4,4,0,11,0,0,255,
     120,3,26
Data 15,241,0,0,3,11,15,15,0,0,255,121,3,5,4,252,
     7,0,4
```

```
Data 4,0,21,0,0,3,26,0,245,4,252,7,0,4,4,0,0,255,
     122,3
Data 26,15,0,241,241,15,0,0,0,255,129,3,26,0,245,
     4,252
Data 7,0,4,4,0,0,18,11,0,15,0,0,7,28,0,3,0,0,14,
     28,0,3
Data 0,0,255,132,3,15,0,7,4,4,7,0,4,252,0,0,18,26,
     0,241
Data 0,0,18,15,252,252,249,0,252,4,0,0,7,28,0,3,0,
     0,14
Data 28,0,3,0,0,255,142,3,11,0,25,4,4,7,0,4,252,0,
     231,0
Data 0,18,26,241,0,0,0,7,43,0,3,0,0,14,43,0,3,0,0,
     255,148
Data 3,15,0,7,4,4,7,0,4,252,0,249,252,252,249,0,
     252,4,0
Data 0,7,28,0,3,0,0,14,28,0,3,0,0,255,153,3,15,0,
     21,4,4
Data 7,0,4,252,0,235,252,252,249,0,252,4,0,0,7,43,
      0,3,0
Data 0,14,43,0,3,0,0,255,154,3,40,0,231,4,252,7,0,
     4,4,0
Data 25,0,0,7,43,0,3,0,0,14,43,0,3,0,0,255,158,3,
     1,0,35
Data 4,4,7,0,4,252,0,250,252,252,253,0,0,0,14,26,
     4,252
Data 0,249,252,252,249,0,0,0,255,255,-1
```

```
' ******************************************
' **                                      **
' **            DEMO-PROGRAMM             **
' **       (c) MAXON Computer GmbH        **
' ******************************************
'
@Text_install
A$="ST-COMPUTER"+Chr$(0)
Xs%=100
Ys%=100
Hidem
'
Do
  X_neu%=Mousex
  Y_neu%=Mousey
  If X_neu%<>X% Or Y_neu%<>Y% Or Xs_alt%<>Xs% Or
                Ys_alt%<>Ys% Or Alpha<>Alpha_alt
    '
    ' TEXT AN ALTER POSITION LÖSCHEN
    '
    @Text_color(0,-1,0)
    @Text_exec(A$,X%,Y%)
    '
    ' TEXT AN DIE NEUE POSITION SCHREIBEN
    '
    @Text_parameter(Alpha,0,Xs%,Ys%)
    @Text_color(1,-1,0)
    @Text_exec(A$,X_neu%,Y_neu%)
    '
    X%=X_neu%
    Y%=Y_neu%
    Xs_alt%=Xs%
    Ys_alt%=Ys%
    Alpha_alt=Alpha
  Endif
  '
  If Mousek=1
    Add Xs%,7
    Add Ys%,7
    Xs%=Min(600,Xs%)
    Ys%=Min(600,Ys%)
  Else
    If Mousek=2
      Sub Xs%,7
      Sub Ys%,7
      Xs%=Max(0,Xs%)
      Ys%=Max(0,Ys%)
    Endif
  Endif
  '
  B$=Inkey$
  If B$="+"
    Add Alpha,15/180*Pi
  Else
    If B$="-"
      Sub Alpha,15/180*Pi
    Endif
  Endif
  '
Loop
```

Text-Scrolling im GEM-Fenster

Andreas Hollmann

Das folgende Listing zeigt, wie man eine beliebige ASCII-Text-Datei lädt und diese in einem einfachen GEM-Fenster vertikal scrollt. Die Routine lässt sich in belibige Programme einbinden und bietet besonders bei Programminfos einen besonderen Reiz, da man hier ja mit Showeffekten glänzen darf.

Zuerst werden einige Initialisierungen durchgeführt: Die maximal zulässige Zeilenanzahl ist durch die Anzahl der Elemente des Stringfelds *Zl$()* begrenzt, hier wurde 1000 gewählt, was für die meisten ASCII-Texte ausreichen dürfte. Man könnte zur variablen Felddimensionierung natürlich auch die Häufigkeit des ASCII-Zeichens Nr.13 (=Carriage return) feststellen, das zusammen mit dem ASCII-Zeichen Nr.10 (=Line feed) ein Zeilenende markiert. Dessen Anzahl entspräche dann der Zeilenanzahl der ASCII-Datei. Mit diesem Wert könnte dann das Stringfeld variabel dimensioniert werden. Das Feld wird zunächst mit 74 Leerzeichen pro Feldelement gefüllt, soviele Zeichen pro Zeile haben im Fenster Platz. In diese 'Leerzeilen' werden die Zeilen der ASCII-Datei später linksbündig eingepaßt. Die folgenden Programmzeilen dienen der Fensterverwaltung, der Übersichtlichkeit wegen bekommen die benutzten Window-Messages Namen. Nach der Auswahl einer ASCII-Datei wird diese zeilenweise in das Feld *Zl$()* geladen. Für das Scrollen ist es wichtig, daß jede Zeile die max. Länge von 74 Zeichen hat, sonst wird die alte Zeile nicht komplett überschrieben und rechts bleiben Textreste stehen. Aus diesem Grund werden die Zeilen mit dem Befehl *LSET* linksbündig in das Stringfeld eingepaßt. Mit *Wind_Create(...)* werden das Aussehen und die max. Größe des Fensters festgelegt. Der Wert '211' = binär 11010011 ist eine Bitmaske. Durch das Setzen der einzelnen Bits werden die entsprechenden Bedienungselemente des Fensters eingeschaltet. Es sind maximal 12 Bits möglich:

Titelzeile	%1
Closer	%10
Fuller	%100
Mover	%1000
Infozeile	%10000
Sizer	%100000
Uparrow	%1000000
Dnarrow	%10000000
Vslide	%100000000
Lfarrow	%1000000000
Rtarrow	%10000000000
Hslide	%100000000000

Das hier benutzte Fenster besteht nur aus Titelzeile, Infozeile, Uparrow, Dnarrow und Closer. In den letzen Parameter schreibt GEM das sogenannte *Window-Handle*. Das ist die Kennziffer des Fensters, damit beim Arbeiten mit mehreren Fenstern jedes einzelne angesprochen werden kann. Das Desktop erhält das Handle 0, alle weiteren Fenster die Handles 1-7. Sollte kein Fenster mehr verfügbar sein (AES verwaltet bis zu 8 Fenster), wird der Wert -1 zurückgeliefert.

Die Prozedur *Show_Window* setzt Titel- und Infozeile *(Wind_Set(...))*, das Fenster wird mit *Wind_Open(...)* geöffnet und der Fensterinhalt gelöscht. Jetzt wird der Textanfang im Fenster ausgegeben und die Kontrolle an das AES übergeben. In der 'REPEAT... UNTIL'-Schleife werden die Fensterelemente abgefragt. Wurde *Uparrow* oder *Dnarrow* angeklickt, wird zu den Scrollroutinen verzweigt, wurde der Closer angeklickt, wird das Fenster geschlossen. Jetzt kommt der Kern der Sache, die Scroll-Routinen. Im Fenster werden 20 Textzeilen dargestellt. Um nun z.B. nach oben zu scrollen, müssen die oberen 19 Zeilen nach unten geschoben und die neue Zeile 1 ausgegeben werden. Die Ausgabe der neuen Zeile 1 sollte mit einer einfachen Textausgabe wohl keine Probleme bereiten. Und das Verschieben der oberen 19 Zeilen erledigt man mit dem fixen *BITBLT*-Befehl. - Denkste ! - Der Text schlingert in wellenförmigen Bewegungen ("An der Nordseeküste...") derart lustlos über den Bildschirm, daß man nur mit Mühe die aufkommende Übelkeit unterdrücken kann. Erst wenn man den Blitter einschaltet (hat aber nicht jeder), kann man damit leben. Also begab ich mich auf die Suche nach einer

schnelleren Möglichkeit, den Bildschirm durch die Gegend zu schieben und fand diese in Gestalt des (übrigens nicht im Handbuch dokumentierten) Befehls *MEMORY_MOVE (B)*. Mit diesem Befehl kann man recht schnell Speicherbereiche kopieren; eigentlich müßte er *MEMORY_COPY* heißen, da der Speicherbereich ab der Quelladresse ja nicht gelöscht wird. Der Befehl *MEMORY_MOVE <Quelle>, <Anzahl> TO <Ziel>* verschiebt eine Anzahl Bytes von der Quell- zur Zieladresse. Da dieser Befehl intern wortweise arbeitet, müssen alle drei Werte gerade Zahlen sein. Falls dies nicht möglich ist, gibt's den Befehl *MEMORY_MOVEB <Quelle>,<Anzahl> TO <Ziel>*, der intern byteweise arbeitet, dadurch auch aber erheblich langsamer ist. Der Aufbau des Bildschirmspeichers wurde schon öfters in verschiedenen Artikeln und Büchern erklärt, deshalb setze ich ihn als bekannt voraus.

Da am linken Rand ein Stück des Hintergrundes und des Fensterrahmens zu sehen ist, die natürlich nicht verschoben werden sollen, fängt der Scrollbereich ab dem 2. Datenwort einer Pixelzeile an. Die Zeile ist 37 Wörter = 74 Bytes lang, und die rechte Grenze liegt beim 38. Wort. Dieser Bereich wird nun 16 Pixelzeilen (= 1280 Bytes) nach unten kopiert; nachdem man 16 Pixel-Zl. kopiert hat, ist 1 Textzeile verschoben worden. Doch da tritt ein ziemlich fieser Denkfehler auf: Die 1. Textzeile wird in die 2. Textzeile kopiert, die 2. Textzeile wird in die 3. - HALT! In der vorher 2. Textzeile steht ja jetzt die 1. Textzeile! Ergebnis des ersten Scrollversuchs: Auf dem Bildschirm steht 20mal die 1. Textzeile. Der Fehler ist umso gemeiner, da er beim Verschieben in die andere Richtung nicht auftritt. Um diesen Fehler zu verhindern, muß der Schleifenzähler gegen die Verschieberichtung laufen, d.h. beim Verschieben nach unten muß man in der letzten Zeile anfangen und sich nach 'oben arbeiten', beim Verschieben nach oben läuft der Schleifenzähler wie gewohnt von oben nach unten. Mit *MEMORY_MOVE* ist das Scrolling etwas schneller als mit *BITBLT*, doch richtig ab geht's erst, wenn das Programm compiliert wird; das Scrolling erreicht dann etwa die 5fache Geschwindigkeit. Noch etwas schneller geht's in Assembler; ersetzt man die FOR...NEXT-Schleife in den Scrollroutinen durch den *INLINE*-Befehl, so kann man auch im Interpreter schnell scrollen. Damit der Compiler die INLINE-Strings verdaut, darf man nur die Register *d1-d5* und *a0-a2* benutzen. Außerdem darf am Ende der INLINE-Routine kein *RTS* stehen, wie es beim Aufruf mit *CALL* notwendig ist (scheinbar erledigt INLINE das selbst). *REPEAT...UNTIL MOUSEBUT=0* sorgt dafür, daß bei festgehaltener linker Maustaste der Text immer weiter scrollt; da das Betriebssystem bei *up_arrow* und *dn_ arrow* keine festgehaltene Maustaste bemerkt. Erst ab der TOS-Version 1.4 funktioniert das, dort kann die Schleife weggelassen werden. Nachdem der Text um eine Zeile gescrollt worden ist, wird die Variable *Topline* um 1 erhöht/erniedrigt, da ja jetzt eine andere Textzeile in der 1. Zeile steht. Die Nr. der neuen Zeile 1 wird mit *TEXT*... direkt in die Infozeile des Fensters geschrieben. Wer jetzt schreit, daß das keine saubere GEM-Programmierung ist, der kann ja 'mal die Infozeile mit *WIND_SET(...)* aktualisieren (ei, wie das flackert und lange dauert...).

Noch ein paar Bemerkungen zur Ausgabe mit *TEXT*: Solange man die Ausgabe horizontal auf Bytegrenzen legt, d.h. die x-Ordinate ohne Rest durch 8 teilbar ist, und auf Textattribute (kursiv, fett, hell...) verzichtet, ist der Befehl recht schnell. Wer's nicht glaubt, kann ja mal

```
'*********************************************
'*         Text-Scrolling im GEM-Fenster      *
'*  Autor:     Andreas Hollmann, Paderborn    *
'*             (c) MAXON Computer GmbH        *
'*  Sprache:   Omikron.BASIC                  *
'*********************************************
'
DEFINTL "A-Z"
Prg_Init'                    einige Initialisierungen
                             durchführen
F_Select(Load$,Button%L)'   ASCII-Text auswählen
IF Button%L=1 THEN '         'OK' Button gewählt
  Load_Ascii(Load$,F_Len%L)'ASCII-Datei laden
  Appl_Init'                 GEM-Anwendung anmelden
  MOUSEOFF
  Wind_Create(211,0,0,640,400,W1%L)' Fenster
                                      entwerfen
  F_Len$= SPACE$(22)+"Länge: "+ STR$(F_Len%L)+"
                                          Bytes"
  Wf_Name$=" "+Load$+" "'       Inhalt Titelzeile
  Wf_Info$=" Zeile: 1"+F_Len$' Inhalt Infozeile
  Show_Window(W1%L,630,370,Wf_Name$,Wf_Info$)
  Show_Text(Topline%L)'       Textanfang ausgeben
  MOUSEON
  Watch_Mouse(W1%L)'        jetzt kann
                            gescrollt werden
  Appl_Exit'                GEM-Anwendung abmelden
ENDIF
PRINT ""'                   Cursor ausschalten
END '                       Programm beenden
'=============================================
DEF PROC Prg_Init
  PRINT ""'                Cursor ausschalten
    XBIOS (Scr%L,2)'          Bildschirm-
                              Adresse in Scr
  DIM Z1$(1000)'              Platz für 1000 Zeilen
                              reservieren
  FOR Offset%L=0 TO 999'      Feld mit
                              Leerzeichen füllen
    Z1$(Offset%L)= SPACE$(74)
  NEXT Offset%L
  DIM Msg%L(10)'            Platz für Message-Puffer
                            reservieren
  Name_Mem%L= MEMORY(128)'        Platz für
                                  Titelzeile und
  Info_Mem%L=Name_Mem%L+64'       Infozeile
                                  reservieren
  Wm_Closed%L=22'                 Window-Messages
                                  benennen...
  Wm_Arrowed%L=24
  Wa_Upline%L=2
  Wa_Dnline%L=3
  Wf_Name%L=2
  Wf_Info%L=3
  Wf_Workxywh%L=4
RETURN
'---------------------------------------------
DEF PROC F_Select(R Path_File$,R Button%L)
  LOCAL Drive%L,Drive$,Path$,F_Name$,Cut%L
  GEMDOS (Drive%L,25)'         aktuelles Laufwerk
                               ermitteln
  Path$= CHR$(65+Drive%L)+":"' Pfad = aktuelles
                                  Laufwerk
  MOUSEON
  FILESELECT (Path$,F_Name$,Button%L)
  MOUSEOFF
  Cut%L= INSTR( MIRROR$(Path$),".")+1' *.EXT
                                     abschneiden
  Path_File$= LEFT$(Path$,( LEN(Path$)-Cut%L))+
                                         F_Name$
RETURN
'---------------------------------------------
DEF PROC Load_Ascii(Load$,R F_Len%L)'ASCII-
                                     Datei laden
  LOCAL Buffer$
  CLS
  TEXT 240,192,"Datei wird geladen..."
  OPEN "i",1,Load$
  F_Len%L= LOF(1)
  Lines%L=1
  WHILE NOT EOF(1)
    LINE INPUT #1,Buffer$'     1 Zeile in Puffer
                                           laden
    LSET Z1$(Lines%L)=Buffer$'   auf 74 Zeichen
                                 bringen
```
→

die x-Ordinate von 16 auf 17 setzen und ein beliebiges Attribut mit *TEXT STYLE* setzen (Kaffeemaschine einschalten wird empfohlen). Tja und dann muß an dieses Listing natürlich noch die GEMLIB angehängt werden. Es werden jedoch nur einige der GEM-Prozeduren benötigt. Da sich die GEM-Prozeduren teilweise gegenseitig aufrufen, ist das Aussortieren ziemlich nervtötend (eigentlich soll das Programm *GEMSEL.BAS* auf der Omikron.BASIC-Diskette das können, aber bei mir hat es bisher versagt). Nur die folgenden Prozeduren/Funktionen werden benötigt, wobei die Zahlen dahinter die Anzahl der Übergabeparameter angeben (einige Prozeduren mit gleichen Namen unterscheiden sich nur durch die Anzahl der Parameter; keine besonders elegante Lösung...).

So, das war's zum Thema Text-Scrolling in Omikron. BASIC. Wer unbedingt mehr als 74 Zeichen pro Zeile verarbeiten möchte, kann ja eine Routine für horizontales Scrolling mit *Lfarrow* und *Rtarrow* programmieren.

Benötigte Prozeduren:

Appl_Init, Appl_Exit, Evnt_ Mesag(1), Menu_Bar, Form_ Dial(5), Form_Dial(9), Graf_ Handle(5), Graf_ Mouse(1), Wind_Create(6), Wind_Open (5), Wind_ Close(1), Wind_ Delete(1), Wind_Get(6), Wind_Set(4), Wind_Update(1), Rsrc_Free, Vst_ Height(5), V_Clswk, V_ Clsvwk, Vs_Ndc, Patch_ Basic(4).

Benötigte Funktionen:

Wind_Check(2), Rsrc_ Check(1), V_Opn_ Check(1), V_Opnv_ Check(1), Reserved(1).

```
      Lines%L=Lines%L+1'           Zeilen-Anzahl um
                                   1 erhöhen
    WEND
    CLOSE 1
    CLS
  RETURN
  '-------------------------------------------------
  DEF PROC Watch_Mouse(W_Handle%L)'  auf Maus-
                                    Klick warten
    LOCAL I%L,Message$,Msg%L
    REPEAT '    solange in der Schleife bleiben,...
      Evnt_Mesag(Message$)
      FOR I%L=0 TO 7
        Msg%L(I%L)= CVI( MID$(Message$,I%L*2+1))
      NEXT I%L
      IF Msg%L(0)=Wm_Arrowed%L THEN ' Pfeil wurde
                                     angeklickt
        IF Msg%L(4)=Wa_Upline%L THEN 'Pfeil
                                     nach oben
          Scroll_Up(Topline%L)'      1 Zeile
                                     nach oben
        ENDIF
        IF Msg%L(4)=Wa_Dnline%L THEN 'Pfeil
                                      nach unten
          Scroll_Dn(Topline%L)'      1 Zeile
                                     nach unten
        ENDIF
      ENDIF
    UNTIL Msg%L(0)=Wm_Closed%L'...bis Closer
                                 angeklickt wird
    Wind_Close(W_Handle%L)
  RETURN
  '-------------------------------------------------
  DEF PROC Show_Text(R Topline%L)'    Text-Anfang
                                      ausgeben
    LOCAL I%L
    TEXT HEIGHT =13'             normale Textgröße
    TEXT STYLE =0' sonst wird die Textausgabe zu
                   langsam
    FOR I%L=0 TO 9
      TEXT 16,((13-I%L)*16+13),Zl$(10-I%L)
      TEXT 16,((14+I%L)*16+13),Zl$(11+I%L)
    NEXT I%L
    Topline%L=1'                 oberste Zeile ist
                                 Zeile 1
  RETURN
  '-------------------------------------------------
  DEF PROC Scroll_Up(R Topline%L)'
    LOCAL Adr%L,Scr_Dn%L
    MOUSEOFF '             sonst gibt's viele
                           Mauspfeile
    REPEAT '               solange scrollen, ...
      IF Topline%L=1 THEN EXIT '     Text-Anfang
                                     erreicht
      'mit Basic-Befehlen:
      'FOR Adr=29442 TO 5122 STEP -80'nach unten
       schieben
      'MEMORY_MOVE Scr+Adr,74 TO Scr+Adr+1280
      'NEXT Adr
      'oder Assembler:
      INLINE"22790000044E43E9730245E90500303C012F4
             CD9013E48D2013E4CD9013E48EA013E00184C
             D9013E48EA013E00303559004843E9FF6645E
             AFFB051C8FFD6"
      Topline%L=Topline%L-1'     oberste Zeile 1
                                 Zeile höher
      TEXT 16,77,Zl$(Topline%L)' neue oberste
                                 Zeile
      TEXT 64,57, STR$(Topline%L)+"  "'  Zeile
                                       anzeigen
    UNTIL MOUSEBUT =0' ...bis Mausknopf
                       losgelassen wird
    MOUSEON
  RETURN
  '-------------------------------------------------
  DEF PROC Scroll_Dn(R Topline%L)
    LOCAL Adr%L
    MOUSEOFF
    REPEAT '               solange scrollen, ...
      IF Topline%L=Lines%L-20 THEN EXIT ' Text-
       Ende erreicht
      'mit Basic-Befehlen:
      'FOR Adr=6402 TO 30722 STEP 80'  nach oben
       schieben
      'MEMORY_MOVE Scr+Adr,74 TO Scr+Adr-1280
      'NEXT Adr
      'oder Assembler
      INLINE "22790000044E43E9190245E9FB00303
              C012F4CD9013E48D2013E4CD9013E48EA013
              E00184CD9013E48EA013E00303559004843
              E9000645EA005051C8FFD6"
      Topline%L=Topline%L+1'    oberste Zeile 1
                                Zeile tiefer
      TEXT 16,381,Zl$(Topline%L+20)' neue unterste
                                     Zeile
      TEXT 64,57, STR$(Topline%L)+"  "'  Zeile
                                       anzeigen
    UNTIL MOUSEBUT =0' ...bis Mausknopf
                       losgelassen wird
    MOUSEON
  RETURN
  '-------------------------------------------------
  DEF PROC Show_Window(W_Handle%L,W%L,H%L,Name$,
           Info$)'
    LOCAL X%L,Y%L,W%L,H%L
    X%L=(640-W%L)/2'       hor. zentrieren
    Y%L=((400-H%L)/2)+9' ver. zentrieren, +9 wegen
                         Menüzeile
    Wind_Set(W_Handle%L,Wf_Name%L,Name$,
             Name_Mem%L)'Titelzeile
    Wind_Set(W_Handle%L,Wf_Info%L,Info$,
             Info_Mem%L)' Infozeile
    Wind_Open(W_Handle%L,X%L,Y%L,W%L,H%L)'
             Fenster öffnen
    Wind_Get(W_Handle%L,Wf_Workxywh%L,Wx%L,Wy%L,
             Ww%L,Wh%L)'Gr. holen
    MOUSEOFF
    FILL STYLE =0,1'    Füllmuster = weiße Fläche
    PBOX Wx%L-1,Wy%L-1,Ww%L+2,Wh%L+2'
                        Fensterinhalt löschen
    MOUSEON
  RETURN

  '=================================================
  ' Hier müssen die Prozeduren der GEM-Library
    stehen !
  '-------------------------------------------------
```

Kleine Hilfe für

RSC-DIALOGE

in GFA BASIC

T.W. Müller

DIALOGE 'ZU FUSS' DIREKT ZU ERSTELLEN, IST SEHR MÜHSAM. EINE SEHR GROSSE HILFE IST EIN RESOURCE CONSTRUCTION SET, DAS DEM PROGRAMMIERER ERLAUBT, DIALOGE, MENÜS UND DERGLEICHEN SEHR KOMFORTABEL AM BILDSCHIRM ZU ERSTELLEN. EINEN HAKEN HAT ALLERDINGS AUCH DAS SCHÖNSTE RCS: DIE VERWENDUNG VON INDIZIERTEN VARIABLEN (AUCH ARRAYS ODER FELDER GENANNT) BLEIBT DEM PROGRAMMIERER VERSAGT.

Oftmals kommt es vor, daß bei Dialogen mehrere Objekte in ihrer Hauptfunktion gleich sind, aber das Programm mit anderen Parametern versorgen. Nehmen wir an, wir hätten einen Dialog, in den wir Werte eintragen können, z.B. Meßwertepaare. Bei 10 Wertepaaren wären das 20 (!!) Felder, die jeweils mit einer Variablen angesprochen werden. Also müssen 20 Zeilen geschrieben werden, um den Dialog in diesem Punkt auszuwerten, und 20 Zeilen, um die Felder vorzubesetzen - also 40 Zeilen bzw. mehr als eine halbe DIN A4-Seite. Mit indizierten Variablen hätten wir zwei kleine Schleifen von je 3 Zeilen.

Also lohnt es sich in einem solchen Fall, sich Gedanken darüber zu machen, wie die Überführung der Objektnamen in indizierte Variablen am besten und einfachsten zu machen ist.

In GFA BASIC 3.xx ist dieses Problem recht elegant zu lösen, gibt es doch den Befehl *ABSOLUTE*. Anfangs stand ich diesem Befehl äußerst skeptisch gegenüber. Was sollte es denn nützen, wenn ich damit eine Variable an einer bestimmten Adresse ansiedeln würde? Doch ich wurde schnell eines Besseren belehrt: Mit Hilfe von ABSOLUTE ist es möglich, Array-Elementen einen 'Decknamen' zu verpassen. Die Vorgehensweise ist denkbar einfach:

- Array dimensionieren,
- Adresse des untersten Elements bestimmen
- mit ABSOLUTE den Elementen einen zweiten Namen geben.

Ein kleines Beispiel:

```
OPTION BASE 0
DIM x&(5)    adr%=V:x&(0)
ABSOLUTE adr%+0,x0& !X&(0)
ABSOLUTE adr%+2,x1& !X&(1)
ABSOLUTE adr%+4,x2& !X&(2)
                "
                "
ABSOLUTE adr%+10,x5&
!X&(5)
```

Das ist schon alles! Nun hört die Variable *x&(0)* auch auf den Namen x0&. Die Sprünge von 2 kommen zustande, da eine WORD-Variable 2 Bytes lang ist. Das nun folgende Programm nimmt dem Programmierer die Schreibarbeit ab, setzt OPTION BASE 0 voraus. In der Procedure 'rom' sähe das Beispiel so aus:

```
DATA x,0,5
DATA *
```

GFA BASIC hat gegenüber Pascal oder C den Nachteil, keine Konstanten zu kennen, jeder Objektname verbraucht also zwei Bytes Speicherplatz (Verwendung von RSC2. PRG). Würden die Objektnummern in die entsprechenden Array-Elemente kopiert, so verdoppelte sich der Speicherplatz, zuzüglich der Deskriptoren. Durch den Kniff mit ABSOLUTE wird immerhin die Verdoppelung des Speicherplatzes verhindert.

Vorgehensweise in eigenen Programmen

Zunächst werden die Arrays dimensioniert und den Elementen die entsprechenden

Namen zugewiesen. Dann werden die Namen auf ihre Werte gesetzt, ist die Resource einsatzbereit. Den zusätzlichen Speicherplatz, der benötigt wird, belegen nur die Felddeskriptoren.

Beim Programm selbst wurde auf jeden 'Schnickschnack' wie z.B. eine Fileselektorbox zur Eingabe des Dateinamens verzichtet. ads läßt sich ja leicht selbst einbauen. Die Namen für Prozedur und Datei müssen also direkt eingesetzt werden.

Sämtliche DIM- und ABSOLUTE-Anweisungen werden vom Programm in eine Prozedur, mit *adr%* als lokaler Variablen, verfrachtet. Das hat den Vorteil, daß die gesamte Procedure mit *HELP* auf eine Editorzeile zusammengeschrumpft werden kann und nicht mehr stört.

In Listing 1 ist das gesamte Programm, das so kurz wie möglich gehalten wurde, in Listing 2 sehen Sie einen Treiber und die Prozedur, die mit dem Programm aus Listing 1 erstellt wurde.

Literatur:

Engels/Görgens.
GFA BASIC 3.0 Handbuch.
GFA Systemtechnik GmbH. 1988

```
' (c) MAXON Computer GmbH
'
RESTORE
DIM d$(200,3)
DIM k|(200,1)
i|=0
REPEAT
  INC i|
  FOR k|=1 TO 3
    READ d$(i|,k|)
    EXIT IF d$(i|,k|)="*"
  NEXT k|
  IF d$(i|,1)="*"
    INC k|
  ENDIF
UNTIL d$(i|,k|-1)="*"
n|=i|-1
'
FOR i|=1 TO n|
  k|(i|,0)=VAL(d$(i|,2))
  k|(i|,1)=VAL(d$(i|,3))
NEXT i|
'
OPEN "O",#1,"comput.LST"
PRINT #1,"PRO test"
PRINT #1,"LOC adr%"
FOR i|=1 TO n|
  PRINT #1,"DIM "+d$(i|,1)+"&(";k|(i|,1);")"
NEXT i|
PRINT #1,"'"
FOR i|=1 TO n|
  PRINT #1,"adr%=V:"+d$(i|,1)+"&(0)"
  FOR k|=k|(i|,0) TO k|(i|,1)
    PRINT #1,"ABSOLUTE "+d$(i|,1);k|;"&,adr%+";k|*
              2;" !";d$(i|,1);"&(";k|;")"
  NEXT k|
NEXT i|
PRINT #1,"RET"
CLOSE
'
PROCEDURE rom
  DATA skal,1,14
  DATA lin,1,14
  DATA *
RETURN
```

```
test
PRINT skal12&
skal&(12)=-123
PRINT skal12&
'
PRINT
'
PRINT lin&(3)
lin3&=555
PRINT lin&(3)
END
'
PROCEDURE test
  LOCAL adr%
  DIM skal&(14)
  DIM lin&(14)
  '
  adr%=V:skal&(0)
  ABSOLUTE skal1&,adr%+2 !skal&(1)
  ABSOLUTE skal2&,adr%+4 !skal&(2)
  ABSOLUTE skal3&,adr%+6 !skal&(3)
  ABSOLUTE skal4&,adr%+8 !skal&(4)
  ABSOLUTE skal5&,adr%+10 !skal&(5)
  ABSOLUTE skal6&,adr%+12 !skal&(6)
  ABSOLUTE skal7&,adr%+14 !skal&(7)
  ABSOLUTE skal8&,adr%+16 !skal&(8)
  ABSOLUTE skal9&,adr%+18 !skal&(9)
  ABSOLUTE skal10&,adr%+20 !skal&(10)
  ABSOLUTE skal11&,adr%+22 !skal&(11)
  ABSOLUTE skal12&,adr%+24 !skal&(12)
  ABSOLUTE skal13&,adr%+26 !skal&(13)
  ABSOLUTE skal14&,adr%+28 !skal&(14)
  adr%=V:lin&(0)
  ABSOLUTE lin1&,adr%+2 !lin&(1)
  ABSOLUTE lin2&,adr%+4 !lin&(2)
  ABSOLUTE lin3&,adr%+6 !lin&(3)
  ABSOLUTE lin4&,adr%+8 !lin&(4)
  ABSOLUTE lin5&,adr%+10 !lin&(5)
  ABSOLUTE lin6&,adr%+12 !lin&(6)
  ABSOLUTE lin7&,adr%+14 !lin&(7)
  ABSOLUTE lin8&,adr%+16 !lin&(8)
  ABSOLUTE lin9&,adr%+18 !lin&(9)
  ABSOLUTE lin10&,adr%+20 !lin&(10)
  ABSOLUTE lin11&,adr%+22 !lin&(11)
  ABSOLUTE lin12&,adr%+24 !lin&(12)
  ABSOLUTE lin13&,adr%+26 !lin&(13)
  ABSOLUTE lin14&,adr%+28 !lin&(14)
RETURN
```

PREVIEW

Hilfsprogramm zur Darstellung von Texten

Dr. Lutz-R. Frank

BEI DER ARBEIT MIT VERSCHIEDENEN EDITOREN, SEI ES NUN TEMPUS ODER 1ST_WORDPLUS, HAT MIR SCHON DES ÖFTEREN EINE MÖGLICHKEIT GEFEHLT, MEINE "MACHWERKE" SCHON EINMAL IM VORAUS ANZUSEHEN, OHNE GLEICH DEN DRUCKER ANZUWERFEN UND EVENTUELL MAL WIEDER EINEN SCHLECHTEN AUSDRUCK ZU ERHALTEN.

Durch die diversen DTP-Programme, die so nach und nach in den verschiedenen Zeitschriften getestet wurden, kam mir die Idee, ein Programm zu schreiben, mit dem ich mir meine Textfiles auch einmal vor dem Ausdrucken im endgültigen Druckbild ansehen konnte. Damit dieses Programm auch aus anderen Programmen wie z.B. 1st_Word heraus zu benutzen ist, lag die Programmierung als Accessory nahe.

Um das Programm universell nutzbar zu machen, habe ich in diesem Programm den in dieser Zeitschrift bereits beschriebenen Trick mit der externen Variablen _app benutzt (das geht nicht nur aus Turbo-C, wie im entsprechenden Handbuch beschrieben - nein, auch mit Megamax Laser-C kann man diesen Trick anwenden). Wenn man das Accessory in PRG umbenennt, kann man es ganz normal aus dem Desktop starten, verliert dann allerdings den ACC-typischen Vorteil. Weiter benötigt das Programm eine Resourcefile, die jedoch nur aus einer beinahe bildschirmgroßen Box, in der nur ein Exit-Button vorhanden sein muß, besteht. Der Inhalt des zugehörigen Header-Files:

```
#define TREE 0
#define EXITALL 1
```

Die verschiedenen Routinen:

main(): Nach dem Anmelden der Applikation und Öffnen der Workstation mit v_opnvwk() wird das zugehörige Resourcefile geladen. Wenn es nicht gefunden werden kann, wird das Programm abgebrochen. Anschließend wird die bereits erwähnte Variable _app abgefragt und der eigentliche Job entweder direkt, oder erst nach einem AC_OPEN-Event ausgeführt.

job(): In dieser Routine wird der Bildschirm gerettet und die Dialogbox dargestellt, der Name der anzuzeigenden ASCII-Datei abgefragt und dieser Name mit draw_melde(cx,cy,name) in eine Statuszeile geschrieben. Schließlich wird die Datei geöffnet, das erste 'Blatt' gezeichnet und der Inhalt der Datei Zeichen für Zeichen eingelesen. Die eingelesen Zeichen werden geprüft, ob es Steuerzeichen (Linefeed, Formfeed etc.) sind und allenfalls als einzelne Pixel auf dem 'Blatt' dargestellt. Wenn notwendig, werden weitere 'Blätter' zur Verfügung gestellt. Alle zehn 'Blätter' muß durch Antippen irgendeiner Taste die unterbrochene Ausgabe der Datei weitergeschaltet werden. Wenn die gesamte Datei dargestellt ist, wartet das Programm auf den Druck auf den Exit-Button, der Bildschirm wird restauriert und das Programm verlassen.

hole_file(filename): Einlesen des Dateinamens mit der Fileselectbox.

draw_seite(cx,cy) und neues_blatt(): Ein leeres 'Blatt' wird die VDI-Funktionen vsf_color(handle,farbe), vsf_interior(handle,typ), pt_set (points[0],cx,cy), v_fillarea(handle,count,points) und vsf_style(handle,typ) dargestellt.

draw_melde(): Eine Statuszeile wird mit den bereits erwähnten VDI-Grafikfunktionen gezeichnet und der vollständige Dateiname wird hier eingetragen.

set_pixel(cx,cy): Über den Umweg pt_set(points[0],cx,cy) und v_pline(handle, count, points) wird an der übergebenen Position ein Pixel gesetzt (Das geht bestimmt noch einfacher - mir wollte hier aber nichts besseres einfallen...).

check_word(): Diese Routine ist der Versuch, 1st_Word-Dateien an ihrem typischen Anfang zu erkennen und das immer vor den Texten stehende Lineal auszublenden. Dazu sucht das Programm in den

ersten 250 Byte nach dem 1st_Word typischen Steuerzeichen des Textanfangs. Wird es nicht fündig, spult es wieder zum Anfang der Datei zurück (Fseek(fd,0L,0);).

So unbeholfen diese Routine auch aussieht, ihren Zweck erfüllt sie bei mir bisher tadellos. Damit wünsche ich viel Spaß beim Abtippen und Ausprobieren des Programmes. Es sollte eigentlich mit allen ST-Konfigurationen zusammenarbeiten. Bei geladenem GDOS (Vers. 1.1) kann es allerdings, wie ich festgestellt habe, zu leichten Problemen kommen. Warum, ist mir allerdings noch nicht klar. Mit Hilfe der im Programmkopf gesetzten Konstanten MAX_ZEILE und MAX_SPALTE sowie MAX_1stWORD und MAX_NORMAL kann sich schliesslich jeder das Programm auf seinen Drucker anpassen (Wieviele Zeichen gehen in eine Zeile, wieviele Zeilen auf ein Blatt). Vorsichtig sein sollte man mit zu vielen Steuerzeichen in einer Zeile. Da das Programm auf diese Steuerzeichen testet, kann es bei zu vielen derartigen Umschaltungen (kursiv, unterstrichen etc.) speziell bei 1st_Word-Dateien zu unschönen Differenzen kommen. Ein mäßiger Gebrauch macht aber keine Probleme.

```
/*


           PREVIEW.C

              in

       Megamax - Laser C
           03.04.89
         Lutz-R. Frank
            V. 1.1
   (c) MAXON Computer GmbH

*/

#include <stdio.h>
#include <osbind.h>
#include <gemdefs.h>
#include <obdefs.h>
#include <string.h>

#include "preview.h"

#define MAX_SPALT 85
#define CX_START  28
#define CY_START  35
#define ACC                    1

extern gl_apid;
extern int _app;

int  MAX_ZEILE=60; /*  hier können die
                       gewünschten Längen  */
int  MAX_1stWORD=54; /*  für 1stWORD und normale
                         Texte ein-  */
int  MAX_NORMAL=60;  /*  geben werden ... */
int  WORD=10;

char fname[128];
char buffer;
char old;
int  exit_bttn=1;
int  fd;
int  wid;
int  ret;
int  xd,yd,wd,hd,
     start_obj,
  exit_obj;

int  pxarray[128],
  contrl[12],
  intin[128],
  ptsin[128],
          intout[128],
  ptsout[128];

int  handle,
  msgbuf[8],
  app_id,
  menu_id;

int  work_in[12];
int  work_out[57];

OBJECT *tree;

```

```

main()
{
 int i;

 app_id=appl_init();
 for(i=0; i<10; work_in[i++]=1);
    work_in[10]=0;
    v_opnvwk (work_in, &handle, work_out);
    if(! rsrc_load("preview.rsc"))
    {
         form_alert(1,
         "[3][    Fataler Fehler | Ich kann das
                  RSC-File |        nicht finden !
                                    ][ Sorry ]");

         v_clsvwk(handle);
         appl_exit();
         return;
 }
 if(!_app)
 {
         menu_id=menu_register(gl_apid,"  Text
                                  Preview ");
         while(1)
         {
                 evnt_mesag(msgbuf);
                 if(msgbuf[0]==AC_OPEN)
                 {
                                 job();
                         }
                 }
 }
 else
 {
         job();
         rsrc_free();
 }
}

/* ------------------------------------------ */

job()
{
 int cx,cy;
 int cx1,cy1;
 int xmin,ymin,xmax,ymax,x,y,b,h;
 int breite,hoehe;
 int zeile=0,
     spalte=0,
     seite=0;
 int count=0;

 wind_get(0,WF_WORKXYWH,&xmin,&ymin,&breite,
          &hoehe);
 xmax=xmin+breite-20;
 ymax=ymin+hoehe-20;
 x=20; y=30; b=615; h=360;
 wid=wind_create(0,xmin,ymin,xmax,ymax);
 wind_update(1);
 wind_open(wid,x,y,b,h);
 wind_update(0);
 rsrc_gaddr(R_TREE,TREE,&tree);
    form_center(tree,&xd,&yd,&wd,&hd);
 form_dial(0,0,0,0,0,xd,yd,wd,hd);
```

→

```
    objc_draw(tree,TREE,MAX_DEPTH,xd,yd,wd,hd);
    hole_file(fname);
    objc_draw(tree,TREE,MAX_DEPTH,xd,yd,wd,hd);
    draw_melde(165,340,fname);
    cx=CX_START;
    cy=CY_START;
    MAX_ZEILE=MAX_NORMAL;
    WORD=0;
    v_hide_c(handle);
    fd=Fopen(fname,0);
    if(fd>0)
    {
            cx1=cx;
            cy1=cy;
            cx=cx1+10;
            cy=cy1+7;
            check_word();
            draw_seite(cx1,cy1);
            while(Fread(fd,1L,&buffer))
            {
                    cx++;
                    spalte++;
                    if(((int)buffer==10) ||
                    (spalte==MAX_SPALT))
                    {
                            cy=cy+2;
                            cx=cx1+10;
                            zeile++;
                            spalte=0;
                    }
                    if(((int)buffer==12) ||
                    (zeile==MAX_ZEILE) ||
                    (((int)&old==27) &&
                    ((int)buffer==-128)))
                    {
                            seite++;
                            spalte=0;
                            zeile=0;
                            cx1=cx1+120;
                            cx=cx1+10;
                            cy=cy1+7;
                            if(seite==5)
                            {
                                    cx1=CX_START;
                                    cx=cx1+10;
                                    cy1=cy1+150;
                                    cy=cy1+7;
                            }
                            if(seite==10)
                            {
                                    seite=0;
                                    cx1=CX_START;
                                    cy1=CY_START;
                                    cx=cx1+10;
                                    cy=cy1+7;
                                    v_show_c(handle);
                                    gemdos(7);
                                    v_hide_c(handle);
                                    neues_blatt();
                            }
                            draw_seite(cx1,cy1);
                    }
                    if(((int)&old!=27) &&
                      ((int)buffer!=-128))
                    {
                            if(((int)buffer>32) &&
                              ((int)buffer<254))
                            {
                                    set_pixel(cx,cy);
                            }
                    }
                    old=buffer;
            }
    }
    v_show_c(handle);
    Fclose(fd);
    exit_obj=form_do(tree,start_obj);
    objc_change(tree,exit_obj,0,xd,yd,wd,hd,NORMAL+
                SHADOWED+OUTLINED,0);
    form_dial(3,0,0,0,0,xd,yd,wd,hd);
    wind_update(1);
    wind_close(wid);
    wind_delete(wid);
    wind_update(0);
}

/* ------------------------------------------- */

hole_file(file_name) /* Filename der Textdatei */
char *file_name[];                   /* einlesen */
{
 int  i;
 char pfad[64],file[64];
 char drive;

 drive=(char) (Dgetdrv()+65);
 strcpy(pfad,"X:\\*.*");
 pfad[0]=drive;
 file[0]=0;

 fsel_input(pfad,file,&exit_bttn);
 i=0;
 while(pfad[i]) i++;
 while(i && pfad[i] != ':' && pfad[i] != '\\')
                                          i--;

 strcpy(&pfad[i+1],file);
 strcpy(fname,pfad);
}

/* ------------------------------------------- */

draw_seite(x_pos,y_pos)     /* Ein leeres "Blatt"
                                    an der     */
int x_pos, y_pos;           /* x,y-Position
                                 zeichnen      */
{
 int count=5;
 int x_len=100,
     y_len;
 int points [5][2];
 int i;

 y_len=2*MAX_ZEILE+20+WORD;
 x_pos=x_pos+2;
 y_pos=y_pos+2;
 vsf_color(handle,1);
 vsf_interior(handle,1);
 pt_set(points[0],x_pos,y_pos);
 pt_set(points[1],(x_pos+x_len),y_pos);
 pt_set(points[2],(x_pos+x_len),(y_pos+y_len));
 pt_set(points[3],x_pos,(y_pos+y_len));
 pt_set(points[4],x_pos,y_pos);
 v_fillarea(handle,count,points);
 x_pos=x_pos-2;
 y_pos=y_pos-2;
 vsf_style(handle,2);
 vsf_interior(handle,8);
 pt_set(points[0],x_pos,y_pos);
 pt_set(points[1],(x_pos+x_len),y_pos);
 pt_set(points[2],(x_pos+x_len),(y_pos+y_len));
 pt_set(points[3],x_pos,(y_pos+y_len));
 pt_set(points[4],x_pos,y_pos);
 v_fillarea(handle,count,points);
}

/* ------------------------------------------- */

neues_blatt()
{
 int count=5;   /* bei zehn angezeigten Seiten */
 int x_len=610, /* vorm Umblättern den Screen  */
     y_len=300, /* restaurieren                */
     x_pos=20,
     y_pos=30;
 int points [5][2];
 int i;

 vsf_perimeter(handle,0);
 vsf_style(handle,4);
 vsf_interior(handle,2);
 pt_set(points[0],x_pos,y_pos);
 pt_set(points[1],(x_pos+x_len),y_pos);
 pt_set(points[2],(x_pos+x_len),(y_pos+y_len));
 pt_set(points[3],x_pos,(y_pos+y_len));
 pt_set(points[4],x_pos,y_pos);
 v_fillarea(handle,count,points);
 vsf_perimeter(handle,1);
}

/* ------------------------------------------- */
```

→

```
draw_melde(x_pos2,y_pos2,filemelde)
int x_pos2, y_pos2;
char *filemelde;
{
 int count=5;  /* den gewählten Dateinamenin   */
 int x_len=350,/* einer Infobox anzeigen       */
     y_len=40;
 int points [5][2];
 int i,j=0;

 vsf_style(handle,2);
 vsf_interior(handle,8);
 pt_set(points[0],x_pos2,y_pos2);
 pt_set(points[1],(x_pos2+x_len),y_pos2);
 pt_set(points[2],(x_pos2+x_len),(y_pos2+y_len));
 pt_set(points[3],x_pos2,(y_pos2+y_len));
 pt_set(points[4],x_pos2,y_pos2);
 v_fillarea(handle,count,points);
 x_pos2=x_pos2+2;
 y_pos2=y_pos2+2;
 x_len=x_len-4;
 y_len=y_len-4;
 do
 {
         pt_set(points[0],x_pos2,y_pos2);
     pt_set(points[1],(x_pos2+x_len),y_pos2);
     pt_set(points[2],(x_pos2+x_len),(y_pos2+
                                      y_len));
     pt_set(points[3],x_pos2,(y_pos2+y_len));
     pt_set(points[4],x_pos2,y_pos2);
     v_fillarea(handle,count,points);
     j++;
     x_pos2++;
     y_pos2++;
     x_len=x_len-2;
     y_len=y_len-2;
 } while(j!=2);
 v_gtext(handle,x_pos2+9,y_pos2+22,filemelde);
}

/* ------------------------------------------- */

set_pixel(cx,cy)
int cx, cy;
{
 int points[2][2];
 int count=2;

 vsf_color(1);
 cy=cy+WORD;
 pt_set(points[0],cx,cy);
 pt_set(points[1],cx,cy);
 v_pline(handle,count,points);
}

/* ------------------------------------------- */

check_word()
{
 int count=0;

 if(Fread(fd,1L,&buffer)>0)
 {
         Fread(fd,1L,&buffer);
         if((int)buffer==48)
         {
                 do
                 {
                         count++;
                         if(count==250) break;
                         if((int)buffer==-128)
                           break;
                         Fread(fd,1L,&buffer);
                 } while((int)buffer!=-128);
                 MAX_ZEILE=MAX_1stWORD;
                 WORD=10;
         }
         else
         {
                 Fseek(fd,0L,0);
         }
 }
}

/* ------------------------------------------- */
```



Einzelheft- u. Disketten Service

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Heim Verlag
Heidelberger Landstr. 194
6100 Darmstadt 13

PD Bestellung

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte freimachen

MAXON Computer GmbH
ST-Computer Redaktion
Industriestraße 26
6236 Eschborn

Kleinanzeigen

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Heim Verlag
Heidelberger Landstr. 194
6100 Darmstadt 13



ABO

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte freimachen

Heim Verlag
Heidelberger Landstr. 194
6100 Darmstadt 13



PUBLIC DOMAIN SERVICE

Ich bestelle folgende PD-Disketten:
(Siehe PD Service in dieser Ausgabe)

____

____

____

____

____

Zahlung erfolgt:

☐ per Scheck

☐ per Nachnahme

Je Diskette fügen Sie bitte einen Betrag von DM 10,– bei, für Porto und Verpackung je Sendung DM 5,– (Ausland DM 10,–)

Datum ______________ Unterschrift ______________



Einzelheft- u. Monatsdisketten Bestellung

ST-Computer können Sie direkt beim HEIM-VERLAG zum Einzelheft-Preis von DM 6,— (1987) DM 7,— (1988) nachbestellen. Bearbeitung nur gegen beigefügten Scheck über den entsprechenden Betrag (keine Überweisung).

Jan.	Febr.	März	April	Mai	Juni	Juli	Aug./Sept.	Okt.	Nov.	Dez.	
											1987 = DM
											1988 = DM
											1989 = DM

+ Gebühr für Porto und Verpackung

(1 Heft DM 2,–, ab 2 Hefte DM 5,–)

☐ Scheck in Höhe zus. DM liegt bei

Disketten Service

Alle Programme, die in ST-Computer veröffentlicht wurden, sind auf Disketten erhältlich. Die Disketten enthalten die Programme von jeweils 2 ST-Computer-Ausgaben. Bestellen Sie durch ankreuzen die gewünschten Disketten (* Monate mit Stern DM 18,—)

* Preis je Diskette 28,— DM	Jan./Febr.	März/Apr.	Mai/Juni	Juli/Aug.*	Sept./Okt.	Nov./Dez.	Jan./Febr.	März/Apr.	Mai/Juni
	87	87	87	87	87	87	88	88	88
	Juli* / Aug./Sept./Okt.	Nov./Dez.	Jan./Febr.	März/Apr.	Mai/Juni	Juli/Aug.	Sept./Okt.	Nov./Dez.	
	88 / 88	88	89	89	89	89	89	89	

Lieferung: gegen beigefügten Scheck zuzügl. 5,— DM Versandkosten, unabhängig der bestellten Menge.

Abonnement

Ja, bitte senden Sie mir die ATARI-Computer Fachzeitschrift ab ______________
für mindestens 1 Jahr (11 Hefte) zum ermäßigten Preis von jährlich DM 70,– frei Haus.
(Ausland: Nur gegen Scheck-Voreinsendung DM 90,– Normalpost. DM 120,– Luftpost)
Der Bezugszeitraum verlängert sich nur dann um ein Jahr, wenn nicht 6 Wochen vor Ablauf des Abonnements gekündigt wird.

Name ______________

Vorname ______________

Straße/Nr. ______________

PLZ ______ Ort ______________

Datum ______________ Unterschrift ______________

Gewünschte Zahlungsweise bitte ankreuzen

☐ Bequem und bargeldlos durch Bankeinzug

Konto-Nr. ______________ BLZ ______________

Institut ______________ Ort ______________

☐ Ein Verrechnungsscheck über DM ______ liegt bei.

☐ Vorauskasse per Zahlung auf unser Postscheck-Konto Ffm, BLZ 500 100 60, Kto.-Nr. 5537-602

Diese Vereinbarung kann ich innerhalb von 8 Tagen beim Heim-Verlag, Heidelberger Landstr. 194, 6100 Darmstadt-Eberstadt widerrufen. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs. Ich bestätige die Kenntnisnahme des Widerrufsrechts durch meine 2. Unterschrift.

Datum, 2. Unterschrift ______________



Kleinanzeigen-Auftrag

Bitte veröffentlichen Sie für mich folgende Kleinanzeige in der angekreuzten Rubrik

Biete an ☐ Hardware ☐ Software **Ich suche** ☐ Hardware ☐ Software ☐ **Tausch** ☐ **Kontakte** ☐ Verschiedenes ☐

30 Buchstaben je Standardzeile – incl. Satzzeichen und Wortzwischenräume.
Groß- und Kleinbuchstaben verwenden, fettgedruckte Wörter unterstreichen.

____

____

____

____

____

____

Bearbeitung **nur gegen Vorausscheck** über den entsprechenden Betrag (keine Überweisung)

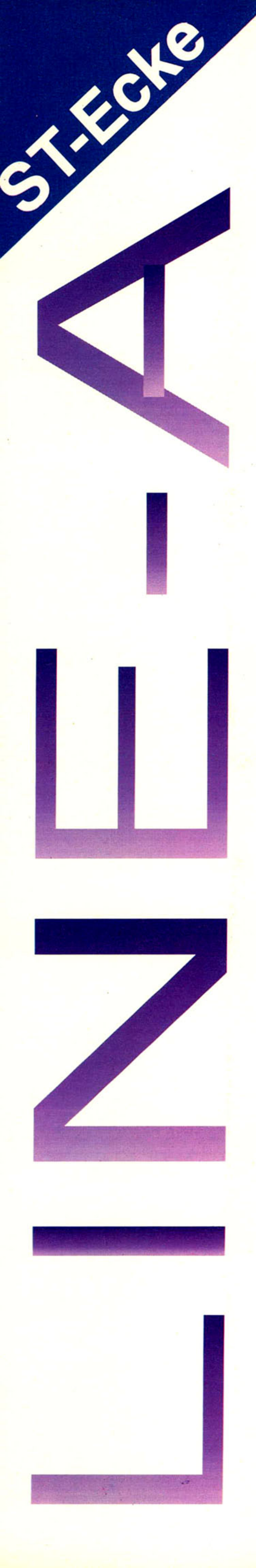
☐ privat = DM 7,– je Zeile
☐ gewerblich = DM 15,– je Zeile
☐ Chiffregebühr = DM 10,–

☐ Scheck über DM ______ ist beigefügt

Bei Angeboten: Ich bestätige, daß ich alle Rechte an den angebotenen Sachen besitze.

Datum ______________ Unterschrift ______________

Absenderangaben auf der Rückseite nicht vergessen





Feuerwerk

Eines Tages saß ich verträumt vor einem Apple Macintosh und schaute so gar nichts ahnend in den leeren, wegen des Bildschirmschoners, dunklen Bildschirm. Was aber war das? Jäh wurde ich aus meinen Tagträumen gerissen, sah ich doch, daß sich etwas auf dem gerade noch so dunklen Bildschirm abspielte. Bei näherem Hinschauen entpuppte sich dieses Etwas als ein schönes kleines Feuerwerk, das den dunklen Alltag des Bildschirmschoners ein wenig aufhellen sollte. Seitdem verfolgte mich der Gedanke, ein solches Feuerwerk auch für den ST zu produzieren - mit dem Erfolg, daß ich mich inzwischen schon öfters vor meinem ST dabei ertappt habe, tagträumend in mein eigenes FIREWORKS zu schauen....

Vorrede

Zugegeben, die Idee stammt nicht von mir, sondern basiert auf einem Programm namens PYRO, welches auf dem Apple Macintosh das Licht der Welt erblickte, trotzdem hat FIREWORKS eine Eigenschaft, auf die es stolz sein kann: es läuft auch auf farbiger Auflösung und das in bis zu 16 Farben - Silvester auf dem Computer. Dieses Programm soll sozusagen den Abschluß unserer LINE-A-Reihe darstellen, da gerade hier LINE-A als die unterste Ebene der Grafikerzeugung optimal zur Anwendung kommt und die höhere Ebene GEM nicht in Anspruch genommen werden kann. Wir werden uns auch ein wenig intensiver als sonst mit Interrupts beschäftigen, da wir unser Feuerwerk aus dem Interrupt heraus starten müssen. Dabei wurde das Feuerwerk

selbst bewußt nicht in Assembler sondern in C programmiert, um Ihnen zu zeigen, wie man C-Routinen sogar im Interrupt benutzen kann. Genug der Vorrede, stürzen wir uns ins Detail.

Das Konzept

Bei der Entwicklung des Programms standen folgende Überlegungen am Anfang: Da FIREWORKS einen Bildschirmschoner darstellt, sollte er automatisch das momentan laufende Programm unterbrechen, sich selbst starten und dann wieder beenden (und damit das unterbrochene Programm fortsetzen), sobald eine Aktion von außen (sprich Maus, Tastatur oder Joystick) getätigt wird. Weiter wäre es ganz schön, wenn man den Bildschirmschoner direkt einschalten oder relativ einfach unterdrücken kann. Diese Optionen sind im dargestellten Listing realisiert worden.

Unterbrechung

Zunächst hat man sich einmal über das System von Interrupts Gedanken zu machen. Stellen Sie sich vor, Ihr Computer hätte nichts anderes zu tun, als Ihr Programm zu bearbeiten. Dies wäre nicht nur langweilig für den Computer, sondern er hätte auch nicht die Zeit, andere wichtige Dinge zu tun. Dazu gibt es sogenannte Unterbrechungen (Interrupts), die den Prozessor dazu veranlassen, daß aktuelle Programm zu verlassen, um kurzzeitig an eine andere Stelle zu springen, dort ein bißchen was zu tun, und dann wieder die Original-Aufgabe (sprich Ihr Programm) weiterzuverarbeiten. Solche Interrupts werden meist hardwaremäßig ausgelöst und geschehen beim Drücken einer Taste und Verschieben der Maus, aber auch durch bestimmte Timer (Zeitgeber-Hardware), die dies zum Beispiel alle fünfzigstel Sekunden hervorrufen.

Einen weiteren Interrupt gibt es in Zusammenhang mit dem Bildschirmaufbau: Der Bildschirm wird zeilenweise aufgebaut, in dem ein Strahl von links nach rechts eine Zeile aufbaut, von rechts nach links abgedunkelt (dunkelgetastet) wird und dann mit der nächsten Zeile beginnt. Die Zeit, in der der Strahl von rechts nach links geht und dabei dunkel ist, nennt man *horizontal blank* (horizontale Dunkeltastung). Diese Zeit ist relativ kurz gegenüber dem sogenannten *vertikal blank* (VBL), der dann auftritt, wenn der Elektronenstrahl, natürlich dunkelgetastet, von der rechten unteren Ecke zur linken

```
/*********************************************************************/
/*                                                                   */
/*                       F I R E W O R K S                           */
/*                                                                   */
/*              entwickelt mit MEGAMAX LASER-C  30.9.89              */
/*                 (c) MAXON Computer GmbH                           */
/*                                                                   */
/*********************************************************************/

 #include <osbind.h>
 #include <gembind.h>
 #include <line_a.h>

 #define kbhit()  (Cconis()==-1)
 #define home()   Bconout(2,0x1b), Bconout(2,'H')
 #define absf(a)  (float)((a>0) ? a:(a)*-1)

 #define ST_L 5                      /* Stern-Länge */
 #define EXPL_MAX 10                 /* maximale Anzahl der Explosionen */
 #define N_EXPL 10                   /* # der Explosionen */
 #define MIN_HOEHE res_y/2.0         /* Mindesthöhe=1/2 der Max-Höhe */
 #define AUSBREITUNG 64              /* Ausbreitung des Explosionen */
 #define MIN_WEITE 64                /* Mindestweite */
 #define STERNBREITE 20

 #define IKBDSYS 32               /* Offset des Vektors */
 #define XBIOS 14                 /* Xbios-Funktionsnummer */
 #define GETREZ    4              /* Xbios' Get_resolution */
 #define SUPEREXEC 0x26           /* Routine in Supervisor ausführen */

 extern int _app;                 /* Zeigt, ob Accessory oder Programm */
 int gl_apid;                     /* Applikationsidentifikation */
 int msgbuff[10];                 /* Ereignis-Buffer */

 long zufall();

 LINE_A *a000(), *linea;
 long random, seed;              /* Globalvariablen für Zufallsgenerator */

 typedef struct                  /* Strukt des Line-A-Musters */
 {
    int *patptr;                 /* Zeiger auf Linienmuster */
    int patmsk;                  /* Anzahl der Linien */
 }PATTERN;

 char screen[32256], *screen_poi;/* Speicher f. Feuerwerkbildschirm */

 char *intro_str[]={"Fireworks","by",
                    "Stefan Höhn",
                    "ST Computer Magazin",
                    "---------",
                    "(c) Maxon Computer GmbH 30.9.'89"};

 static vbl_cntr(), wait_time();    /* allgem. Zugriff */
 void firework(), pause(), set_line_font(), set_line_color();
 float curve();

 main()
 {
    char c, ret, status=1;   /* Status weiP, ob fireworks ein */

    appl_init();             /* Wichtig für AES-Routinen */

    seed = Random()&1023;    /* Anfangswert für Zufallszahl */

    screen_poi=(char*)(((long)screen&~0xffL)+0x100);
    *(long*)wait_time=9000L;        /* 9000 VBLs warten */
    ikbd(0);                        /* Interrupt einhängen */

    if (!_app)                      /* Accessory ? */
    {
      menu_register(gl_apid,"  Fireworks");  /* anmelden */
      while(1)
      {
         evnt_mesag(msgbuff);          /* Auf Ereignis warten */
         if (msgbuff[0]==AC_OPEN)
         {
            ret=form_alert(2-status,"[3][Fireworks (c) S.
                                               Höhn][Aus|Ein]");
            if (ret==1 && status==1)  /* nur aus, wenn an */
            {
               ikbd(1);     /* Interrupt aushängen */
               status=0;
            }
            else if (ret==2 && status==0) /* nur an, wenn aus */
            {
```

→

oberen Ecke wandern muß. Von beiden Blanks kann ein Interrupt ausgelöst werden, allerdings hat man während des horizontal blanks kaum mehr Zeit als für ein paar Befehle, hingegen kann im vertikal blank doch einiges erledigt werden. Welchen Zweck hat es aber, daß man bestimmte Dinge nur im VBL erledigt? Ganz einfach! Stellen Sie sich vor, der Rechner baut gerade den Bildschirm auf und ist dabei in der Mitte des Bildschirms angekommen, sie ändern aber gerade den Bildspeicher. Dann finden Sie bis zur Mitte den alten Bildspeicher, die untere Hälfte gehört aber schon zum neuen Bild, was gar nicht schön aussieht. Sie könnten sich auch vorstellen, daß Sie die Hintergrundfarbe ändern, wodurch der obere Teil eine andere Hintergrundfarbe erhielte als der untere. Jetzt werden ein paar unter Ihnen sagen, 'au fein, dann kann ich ja viele Farben als Hintergrund haben', aber der Nachteil ist, daß die Zeile, in der sich der Farbsprung ereignet, mit großer Wahrscheinlichkeit nicht immer die gleiche sein wird, da ihr Programm nicht mit dem Bildschirm synchronisiert ist. Sie sehen, daß wir den Bildschirmaufbau doch besser im VBL-Interrupt machen.

Geordnetes Durcheinander

Was passiert aber, wenn wir nun aus einem 200 Hz-Interrupt eine Routine aufrufen würden, die länger als eine 200stel Sekunde dauerte? Der Effekt kann recht gemein sein: Nehmen wir an, weitere Interrupts seien zugelassen (man kann sie auch sperren), so würde der Prozessor unsere Routine, die vorher durch einen Interrupt ausgelöst wurde, durch den aktuellen Interrupt ein weiteres Mal unterbrechen und damit auslösen und so fort - daß dies auf Dauer nicht gut gehen kann, ist wohl einzusehen (wer das nicht verstanden hat, sollte noch einmal die letzten Sätze langsam durchdenken). Eine Forderung kann zum Beispiel sein, daß der Interrupt nicht mehr als eine gewisse Zeit dauern darf! Eine weitere Möglichkeit ist die folgende: Wird eine eigene Routine durch einen Interrupt angesprochen, fragt diese zunächst ein Flag ab. Ist dieses Flag gesetzt, so sorgt unsere Routine dafür, daß sie sofort wieder zurückkehrt und somit weiterspringt. Dieses Flag setzen wir nämlich dann, wenn unsere Routine gerade vorher aufgerufen worden war und am Arbeiten ist. Verlassen wir unsere Routine wieder, löschen wir dieses Flag und lassen erst dann wieder zu, daß sie erneut

```
                ikbd(0);     /* Interrupt einhängen */
                status=1;
            }
        }
    }
    else   /* nein, Programm */
    {
        *(long*)wait_time=300L;     /* nur Kurz warten */
        while( Cconis())            /* Tastaturbuffer löschen */
            Crawcin();

        do
        {
            evnt_timer(100,0);          /* Ab und zu einmal */
            home();
            printf("vergangene VBLs: %51d\n",*(long*)vbl_cntr);

            if(kbhit())                      /* Taste gedrückt ? */
                c=gemdos(7);                 /* Ja, abholen */

        }while(c!='q');                      /* Taste q bewirkt Abbruch */
        ikbd(1);                             /* Interrupts ausklinken */

        appl_exit();                         /* bei AES abmelden */
    }
}

ikbd(arg)
char arg;
{
    static do_ikbd(), kbdv_addr(), ikbd_on(), ikbd_off();
    static ende(), new_ikbd(), vbl_int(), vbl_count();
    static ikbd_vec(), semaphore(), vbl_end(), vbl_end1 (),vbl_flag();
    static resolution(), hard_phys(), logbase(), our_linea();
    static vbl_cnt(), vbl_cnt2(), vbl_supress(),fireflag();
    static line_adr(),line_trap(), line_flag(), code();

    asm{
    movem.l A0-A2/D0-D2,-(A7)

    move.w #GETREZ, -(A7) /* Funktion Getrez() */
    trap   #XBIOS         /* Xbios-Aufruf */
    addq.l #2, A7         /* Stack korrigieren */
    move.w D0, resolution /* Auflösung abspeichern */

    move.w #34,-(A7)      /* Funktion Kbdvbase() */
    trap #XBIOS           /* Xbios-Aufruf */
    addq.l #2, A7         /* Stack korrigieren */
    lea    kbdv_addr,A1   /* Adresse für Vektortabelle */
    move.l D0,(A1)        /* Adresse der Vektortabelle */

    lea    ikbd_on, A0    /* Adresse: Einbinde-Routine */
    tst.b  arg(A6)        /* Arg = 0 ? */
    beq     do_ikbd       /* Ja  */

    lea    ikbd_off,A0    /* Adresse: Ausblend-Routine */

do_ikbd:
    pea    (A0)                /* Adresse auf Stack */
    move.w #SUPEREXEC,-(A7)    /* Supervisor */
    trap   #XBIOS              /* XBIOS-Einsprung */
    addq.l #6,A7               /* Stack korrigieren */
    jmp    ende                /* Bis dann... */

ikbd_on:
     dc.w    0xa000               /* Linea-Adresse holen */
    move.l  D0, our_linea       /* speichern */

    movea.l kbdv_addr,A0        /* Tabellenadresse */
    lea     ikbd_vec,A1         /* Sicherungsadresse */
    move.l  IKBDSYS(A0),(A1)    /* alten Vektor sichern */
    lea     new_ikbd,A1         /* neue Routine einklinken */
    move.l  A1,IKBDSYS(A0)

    clr.l   vbl_cntr            /* Anfangswert =0 */
    clr.b   semaphore           /* Semaphore löschen */

    lea     vbl_int, A1    /* Adresse zum Speichern */
    move.l  0x70, (A1)     /* Adresse des 200Hz-Interrupts sichern */
    lea     vbl_count, A0       /* unsere Routine einbauen */
    move.l  A0, 0x70            /* in VBL-Zeiger schreiben */

    /* im folgenden wird der Line-A-Vektor verbogen */
    move.l  0x28, line_adr
  lea     line_trap, A0
```

→

durch einen Folgeinterrupt angesprochen wird. Anschauung: Wir stellen uns einen breiten Gang vor, in dem in der Mitte ein kleiner Raum ist. Durch diesen Raum kann eine Person durch die Vordertür hinein- und die Hintertür hinausgehen. Ist in dem Raum schon jemand drin, müssen nachfolgende Personen um den Raum herum weitergehen. Erst wenn die Person den Raum verläßt, kann eine Folge-Person den Raum betreten und dort tun und lassen was sie möchte.

Man muß halt Prioräten setzen

Interrupts werden in verschiedene Prioritäten eingeteilt, d.h. eine Interruptroutine höherer Priorität kann nicht von einer niedrigerer oder gleicher Priorität unterbrochen werden. Diese Priorität bezeichnet man als Interruptlevel, der von 0 bis 7 durchnumeriert ist. Ein Interrupt auf Level 7 kann also nicht mehr unterbrochen werden. Mißbrauchen wir nun einen Interrupt in der oben beschriebenen Art, daß wir dadurch eine eigene Routine starten und dann weitere Interrupts drumherumleiten, müssen wir dafür sorgen, daß wir selbst den Level heruntersetzen. Stellen wir uns nämlich vor, daß zum Beispiel der VBL auf Level 4 ausgelöst wird, so läuft unsere Routine auf Level 4. Da der nächste VBL auch auf Level 4 kommt, der Prozessor aber momentan auf diesem Level läuft, wird dieser Interrupt erst gar nicht angenommen, denn schon bei gleichen Interruptprioritäten wird ein Interrupt verhindert.

Vorbereitung für's Feuerwerk

Um die Einzelheiten unseres Feuerwerks zu verstehen, wenden wir uns nun dem Listing zu und werden besonders die Assemblerroutine unter die Lupe nehmen. Diese Routine heißt *ikbd()* (was eher historisch zu sehen ist, da sie ursprünglich nur den Tastaturinterrupt verbog) und ist in der Lage alle benutzen Interrupts zu verbiegen oder wieder auszuhängen, je nachdem, welches Argument übergeben wird. Gehen wir schrittweise vor... Zunächst holen wir mit der Routine *Getrez()* des Bios die Auflösung, welche wir uns zwischenspeichern, um später in unserem Feuerwerk darauf zugreifen zu können. Jetzt werden Sie sich fragen, warum wir dies nicht erst dann tun, wenn wir sie brauchen. Bei den Interrupts des ATARI ST gibt es eine wichtige Tatsache zu

```
    move.l  A0, 0x28
    rts
ikbd_off:
    movea.l  kbdv_addr,A0      /* Adresse Keyboard-Tabelle */
    lea      ikbd_vec,A1
    move.l   (A1),IKBDSYS(A0) /* Alten Vektor setzen */
    move.l   vbl_int, 0x70  /* alten 200Hz-Interrupt wieder setzen */
    move.l   line_adr, 0x28   /* line_a wieder herstellen */
    rts
line_trap:
    move.l   2(A7), A1     /* Adresse, an der Line-A-Befehl steht */
    move.w   (A1), code    /* Line-A-Befehl */
    addq.l   #2, A1        /* Zeiger auf Adresse nach Befehl */
    move.l   A1, 2(A7)     /* Diese Adresse auf Stack als Rückadresse */
    st       line_flag             /* merke, daP CPU in Line-A steckt */
    pea      code                  /* Adresse, wo unserer Befehl steht */
    move.w   SR, -(A7)             /* SR nach Stack, wie bei exception */
    move.l   line_adr, -(A7)       /* Adresse des Line-As */
    rts                            /* ab dorthin */
code:
    dc.w 0
    clr.b    line_flag             /* CPU wieder aus Line-A */
    rte
new_ikbd:                          /* Tastaturprozessor angerührt */
    clr.l  vbl_cntr                /* Zähler zurücksetzen */
end_ikbd:
    move.l ikbd_vec,-(A7)          /* durch alten Vektor weiter */
    rts
vbl_count:                         /* VBL-Routine */
     movem.l D0/A0-A1, -(A7)       /* Register retten */

     clr.b  vbl_flag               /* Flag für VBL-Warten löschen */

     tst.b  semaphore              /* Semaphore gesetzt ? */
     bne    vbl_end1               /* JA, unseren VBL umgehen !!!! */

     st     semaphore              /* setze semaphore */

     tst.b  line_flag              /* Interrupt aus Line-A heraus ? */
     bne    vbl_end                /* JA, raus hier */

     tst.w  0x43e                  /* flock testen, Laufwerk aktiv ? */
     bne    vbl_end                /* JA, sofort weiter */

     move.l    our_linea, A0  /* Adresse der linea-Variablen */
                             /* Maus auf untere linke
                                Ecke abfragen */
     move.w    -0x25a(A0), D0 /* X */
     bne       vbl_cnt        /* Dann auch nicht 0,0,
                                normal weiter */
     move.w    -0x258(A0), D0 /* Y = */
     addq      #1, D0         /* um eins erhöhen */
     cmp.w     -4(A0), D0     /* mit Y-Auflösung gleich? */
     bne       vbl_supress  /* Nein! Feuerwerk bei 0,0
                               unterdrücken? */
     tst.b     fireflag     /* Schon Feuerwerk durch fireing
                               corner?*/
     bne       set_wait       /* JA, Schlaf weiter */
     move.w    -0x254(A0), D0    /* Nein, Maustasten testen */
     andi.w    #3, D0            /* beide Tasten ausblenden */
     cmpi.w    #3, D0            /* Beide gedrückt */
     bne       vbl_supress       /* Nein, vielleicht 0,0 ... */
     move.b    #1, fireflag      /* In fireing corner gefahren und
                                    merken */
set_wait:
     move.l    wait_time, vbl_cntr /* JA! */
     bra       vbl_cnt2            /* nächste Abfrage überspringen */
vbl_supress:
     move.w    -0x25a(A0), D0      /* X */
     add.w     -0x258(A0), D0      /* +Y */
     tst.w     D0                  /* Flags setzen */
     beq       vbl_end             /* =0?, dann nicht erhöhen */
vbl_cnt:
     clr.b     fireflag            /* Flag löschen */
vbl_cnt2:
     addq.l    #1, vbl_cntr       /* Timer */
     move.l    wait_time, D0      /* Ansprechzeit */
     cmp.l     vbl_cntr, D0       /* Zeit schon erreicht ? */
```

→

Aller guten Dinge sind vier...

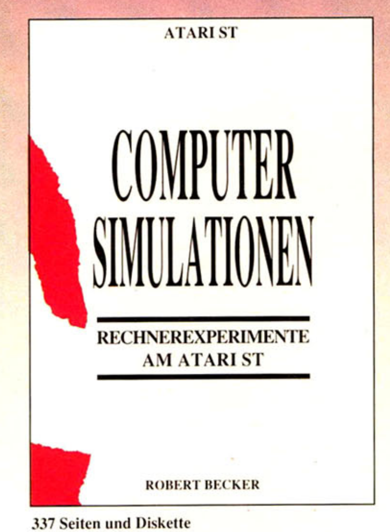
ATARI ST

OMIKRON BASIC 3.0

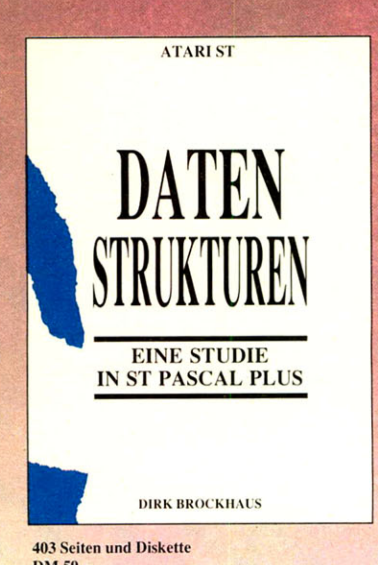
DAS BUCH ZUM HANDBUCH

CLEMENS HOFFMANN

355 Seiten und Diskette
DM 59.-



337 Seiten und Diskette
DM 59,-



403 Seiten und Diskette
DM 59,-

OMIKRON.BASIC 3.0

Das Buch zum Handbuch

Einführung (für den absoluten Neuling):

- Kurze Anleitung in der BASIC-Programmierung.
- Über das Handbuch hinausgehende Beschreibung vieler Befehle, Besonderheiten und Kniffe. Verwendung selbstdefinierter Prozeduren und Funktionen. Viele Beispiele, Aufgaben mit Lösungen.

Für den Aufsteiger, aber auch für den geneigten Anfänger:

- OMIKRON.Sprites-Tücken, Vorteile, Anwendung.
- Overlay-Technik (Auslagern langer Programmteile und Laden bei Gebrauch).
- Grundlagen der strukturierten Programmierung. Schreiben eigener und Verwenden fremder Libraries (Bibliotheks-Funktionen).
- Aufrufe von TOS und GEM im BASIC (GEMLib): endlich die Wahrheit über die GEM-Aufrufe! Dabei wird auch das GEM-Zusatzprogramm GDOS berücksichtigt.
- Sound und Grafik-Programmierung. Grafische Effekte (z.B. die Verwendung mehrerer Grafik-Bildschirme und Zeichnen in nicht sichtbare Bildschirme. Aufbau von Metafiles, IMG-Bildern u.a.
- Aufbau und Verwenden der Menü-Leisten in GEM-Accessories in OMIKRON.BASIC.
- Verwendung der BASIC-internen Multitasking-Befehle.
- Einige Libraries (Turtle-Grafik Erweiterungen und Korrekturen zur GEMLib, usw).
- Erklärung der Befehle der Version 3.0 und ihre Anwendung. Die Feinheiten des Compilers V 2.0.
- Natürlich befinden sich alle Programme und Beispiele auf Diskette!

Computer-Simulationen

Rechnerexperimente am ATARI ST

Die Simulation von Naturvorgängen war und ist eine Stärke von Computern. Wer kennt nicht das alte 'Spiel' Life, das von einer Population unter vorgegebenen Regeln verfolgt wird? Dieses Buch befaßt sich nicht mit Life, aber mit anderen Simulationsvorgängen, die noch viel interessanter sind.

Sie werden in die Lage versetzt, Galaxien kollidieren zu lassen, ohne Ihr Heim zu verlassen oder chemische Reaktionen ablaufen zu lassen, ohne dabei gleich das ganze Haus in die Luft zu sprengen.
Ohne Theorie geht es bei so einem Buch leider nicht, die Experimente sind aber in Form von Listings dabei. Man kann also nach Studie der Theorie das Experiment sofort nachvollziehen. Die Listings sind alle in GFA-BASIC, das wegen seiner Notation einfach zu verstehen und auf andere höhere Programmiersprachen leicht anzupassen ist. Wir haben ein paar Punkte aus dem Inhalt für Sie ausgesucht:

- Simulationsmodelle in den Naturwissenschaften. Die nötigen Begriffe sowie die Gesetzmäßigkeiten werden hier festgelegt. Dieser Teil bildet einen wichtigen Baustein für den gesamten Inhalt.
- Einfache Bahnbewegungen - ein Gasmodell
- Random-Walk-Algorithmen
- Teilchenbewegung in Feldern
- Mehrkörper-Probleme: Das Ende der Theorie
- Doppelsterne und Planetensysteme
- Der radioaktive Zerfall
- Zelluläre Automaten
- Ökologische Modelle

und einiges mehr.

Natürlich liegt eine Diskette bei. Also nur Diskette mit den Programmen einstecken und dann einfach experimentieren!

Datenstrukturen am ST

Eine Studie in Pascal

Datenstrukturen sind ein sehr wichtiger Bestandteil der Informatik. Die Beherrschung dieses Werkzeuges vereinfacht die Programmierung enorm. Dieses Buch von Dirk Brockhaus (Autor einer ähnlichen Serie in der ST COMPUTER) befaßt sich mit verschiedenen Strukturen und Algorithmen. Auf mehr als 400 Seiten mit vielen Bildern sind eine Unmenge von wichtigen Strukturen erklärt und dokumentiert.

Damit der Stoff nicht zu trocken wird, sind die verschiedenen Kapitel durch realistische in CCD-PASCAL plus geschriebene Beispiele verdeutlicht. Diese können aber ohne weiteres in andere PASCAL-Dialekte portiert werden. Wir versuchen, ein paar wichtige Informationen aus dem Inhalt dieses Buches für Sie zusammenzufassen:

- Grundlagen. Einfache Elemente wie Datentypen im allgemeinen werden ausführlich erklärt. Damit wird die Basis für spätere Kapitel geschaffen.
- Komplexe Zahlen. Die in der Mathematik und einigen Ingenieurdisziplinen sehr häufig vorkommenden Datenstrukturen sowie Komplexe Zahlen werden erklärt und anhand einer Anwendung ('Apfelmännchengraphiken') veranschaulicht.
- Suchen und Sortieren der Algorithmen. Verschiedene Methoden des Sortierens werden hier behandelt. Eine Menge Beispiele schaffen mit Sicherheit die richtige Verbindung zwischen Theorie und Praxis.
- Informationsstrukturen. Eine der Hauptaufgaben des Computers ist unbestritten die Verarbeitung von großen Datenmengen gleichen Typs. Die wesentlichen Datenstrukturen, die hinter dieser Aufgabe stehen, werden in diesem Kapitel behandelt.

Und noch viel, viel mehr. Natürlich wird dieses Buch mit einer Diskette geliefert, die alle Beispiele beinhaltet.



Bestellcoupon MAXON Computer GmbH Industriestraße 26 6236 Eschborn Tel.: 06196/481811

Name: ____________________

Vorname: ____________________

Straße: ____________________

Ort: ____________________

Unterschrift: ____________________

Hiermit bestelle ich:

☐ Exemplare von "OMIKRON.BASIC 3.0" mit Diskette für DM 59,00

☐ Exemplare von "Computer-Simulationen" mit Diskette für DM 59,00

☐ Exemplare von "Datenstrukturen" mit Diskette für DM 59,00

Versandkosten: Inland DM 7,50
Ausland DM 10,00
Auslandbestellungen **nur** gegen Vorauskasse
Nachnahme zuzgl. DM 4,00 Nachnahmegebühr.

☐ Vorauskasse
☐ Nachnahme

Vertrieb in der Schweiz: DTZ DataTrade AG Langstrasse 94 Postfach 413 CH-8021 Zürich Tel.: 01/242 80 88 Fax.: 01/291 05 07
Vertrieb in Österreich: Dipl.-Ing. Reinhart Temmel Ges.m.b.H. & Co.KG Markt 109 A-5440 Golling Tel.: 06244/7081-17 Fax.: 06244/7188-3
Vertrieb in Frankreich: AROBACE 2Rue Piemontesis F-75018 Paris Tel: 1/42235044 Fax 1/42545631

beachten: Betriebssystemaufrufe aus dem Interrupt sind strengstens untersagt, das heißt, daß wir ab sofort (fast) alles selbst machen müssen. Dies kann man dadurch ein wenig vermeiden, daß man Aufrufe wie das Holen der Auflösung bei der Initialisierung des Interrupts durchführt und den Wert an einer bestimmten Stelle speichert, die die Interruptserviceroutine kennt. Als nächstes besorgen wir uns die Adresse der Keyboard-Handler-Interrupt-Routine (IKBD). Wie ich oben schon erwähnt habe, lösen Tastatur, Maus und Joystick auch einen Interrupt aus. Diese Einheiten laufen alle über den Tastaturprozessor, der einen Interrupt auslöst und an einer bestimmten Adresse Datenpakete ablegt, die der ST in der IKBD-Routine auswertet. Anders gesagt: Immer dann, wenn ein Datenpaket über diesen Weg ankommt, kann man davon ausgehen, daß die Maus bewegt oder eine Taste gedrückt worden ist. Deshalb merken wir uns die Adresse der Originalroutine und schreiben dann die einer eigenen Routine in einen Zeiger hinein, die nichts anderes macht, als sich zu merken, ob ein Datenpaket kommt und dann in die Originalroutine weiterspringt. Was man sonst noch so alles mit dieser Interruptroutine anstellen kann, finden Sie in der ST-Ecke 'Der ST läßt das Mausen nicht'.

Priviligiert

Leider liegen manche Adressen in einem Speicherbereich, auf den man als normales Programm (unsere Initalisierung läuft ja als ganz normales Programm ab und ist auch in keinster Weise schon ein Interrupt!) nicht zugreifen darf. Dazu kennt der 68000-Prozessor zwei Modi: Den User- und den Supervisor-Modus. Nur im Supervisor-Modus darf auf diese priviligierten Adressen zugegriffen werden, woraus folgt, daß der Programmteil, der unsere Zeiger verbiegen soll, eben dort laufen muß. Leider ist es nicht ganz so einfach vom User- in den Supervisor-Modus zu kommen, aber da hilft uns das XBIOS. Es stellt eine Funktion zur Verfügung (*SUPEREXEC*), der man die Adresse einer Routine angibt, die sie dann im Supervisor ausführt und wieder in den User-Modus zurückkehrt. Schauen wir uns also die beiden Routinen *ikbd_on* und *ikbd_off* an, die, abhängig vom *ikbd()*-Argument *arg*, von *SUPEREXEC* im Supervisor-Modus ausgeführt werden.

Danach ist die Initialisierung der Interrupts abgeschlossen, und es wird mit *jmp ende* an das Ende der Prozedur gesprun-

```
    bpl       vbl_end            /* Nein, ikbd_flag bleibt gelöscht */

    movea.w   #0x8200, A1        /* Hardwareadresse des Shifters */
     dc.w      0x0109              /* movep.w 1(A1), D0 */
     dc.w      0x1

     move.w    SR, -(A7)           /* Statusregister retten */
     move.w    #0x2300, SR         /* Interrupt-Level auf 3 */

     move.w    D0, hard_phys       /* 2 Bytes Physbase abspeichern */

     move.l    0x44e, logbase      /* Logbase sichern */
     move.l    screen_poi, A0      /* Speicherbereich für Feuerwerk */
     move.l    A0, 0x44e           /* logische Adresse setzen */
     move.l    A0, D0
     lsr.l     #8, D0     /* ein Byte nach unten */
     dc.w      0x0189     /* movep.w  D0, 1(A1). physbase schreiben */
     dc.w      0x1

     movem.l   A0-A6/D0-D7,-(A7)/* Besser mal alle
                                   Register retten...*/
     move.w    resolution, -(A7)
     jsr       firework            /* Silvester.... */
     addq.l    #2, A7
     movem.l   (A7)+,A0-A6/D0-D7   /* ...und wieder zurückschreiben */

     st        vbl_flag            /* VBL-Flag setzen */

wait_for_vbl:
     tst.b     vbl_flag            /* immernoch gesetzt ? */
     bne       wait_for_vbl        /* Ja, warten */

     move.w    (A7)+, SR           /* SR retten */

     move.w    hard_phys, D0       /* Physikalische Adresse in */
     movea.w   #0x8200, A1         /* Hardwareadresse des Shifters */
     dc.w      0x0189              /* movep.w  D0, 1(A1). physbase
                                      schreiben */
     dc.w      0x1
     move.l    logbase, 0x44e      /* logbase zurückschreiben */

     clr.l     vbl_cntr            /* Timer rücksetzen */

vbl_end:
     clr.b     semaphore      /* unsere VBL-Routine wieder freigeben */
vbl_end1:
    movem.l (A7)+, D0/A0-A1        /* Register wieder holen */
    move.l vbl_int, -(A7)          /* alten Vektor holen */
    rts

wait_time:      dc.l 0    /* Ansprechzeit in VBL-Einheiten */
semaphore:      dc.w 0    /* VBL-Verriegelung */
resolution:     dc.w 0    /* gespeicherte Auflösung */
vbl_int:        dc.l 0    /* alte VBL-Adresse */
our_linea:      dc.l 0    /* Adresse der linea-Variablen */
ikbd_vec:       dc.l 0    /* alter ikbdsys-Vektor */
kbdv_addr:      dc.l 0    /* Vektor-Tabelle */
vbl_cntr:       dc.l 1    /* Anzahl der vergangenen Millisekunden */
logbase:        dc.l 0    /* Logische Bildschrimadresse */
line_adr:       dc.l 0    /* Adresse der Line-A-Routine */
line_flag:      dc.w 0    /* Merker, ob CPU in Line-A steckt */
hard_phys:      dc.w 0    /* Physikalische Bildschirmadresse */
fireflag:       dc.w 0    /* für die Maus ->sleeping corner */
vbl_flag:       dc.w 0    /* 'auf nächsten VBL warten'-Flag */
ende:
  movem.l (A7)+, A0-A2/D0-D2      /* alles gerettete wieder holen */
   }
}

/* Intro gibt den Anfangstext mittig postiert aus */
intro(res_x, res_y, font)  /* Auflösung x,y, welchen font */
int res_x, res_y;
FONT_HDR *font;
{
   int len, ch_dist, i, x, y, x_start, text_nr;

   for (text_nr=0; text_nr<6; text_nr++)  /* 6 Zeilen Text */
   {
        len=strlen(intro_str[text_nr]);  /* Textzeilenlänge */

      if (!text_nr)      /* bei erster Zeile die */
      {                  /* Buchstaben schön auseinanderziehen */
        ch_dist=(res_x/2-font->max_char_width)/(len-1);
        x_start=res_x/4;
      }
      else               /* sonst Text nur mitteln */
```

→

gen. Alles, was dazwischen liegt, sind Interrupt-Routinen und Anhängsel, aber soweit sind wir nicht - bleiben wir zunächst bei *ikbd_on*. Wie Sie sicherlich noch aus den vorherigen Folgen wissen, muß man, um Line-A zu benutzen, die Adresse der sogenannten Line-A-Variablen wissen. Diese Adresse bekommt man dadurch heraus, daß man den Maschinenbefehl *A000* ausführt, worauf das Line-A unter anderem im Datenregister *D0* die Adresse dieser Variablen zurückgibt. Die Line-A-Variablen werden wir später benutzen. Im folgenden Schritt holen wir uns aus der Keyboard-Handler-Tabelle die Originaladresse der Keyboard-Handler-Interruptserviceroutine, merken sie uns (*ikbd_vec*) und schreiben die Adresse unserer eigenen Routine (*new_ikbd*) hinein. Ab diesem Moment wird jedes Datenpaket erst durch unsere Routine geschleust. Schauen wir uns diese Routine an, erkennen wir, daß sie nichts anderes macht als den Inhalt einer bestimmten Adresse zu löschen (was übrigens ein Zeitzähler ist, aber dazu gleich mehr) und dann sofort in die Originalroutine weiterspringt. Folgen wir weiter der Spur des ikbd_on: Wir initialisieren ein paar Adressen (Variablen), die wir in der eigentlichen VBL-Interrupt-Routine benötigen und die auf einem definierten Wert stehen sollen.

Als erstes wird der Zeitzähler auf Anfang gesetzt. Dieser Zeitzähler wird später in unserer eigenen VBL-Interrupt-Routine bei jedem Aufruf um eins erhöht, worauf unsere VBL-Routine wieder verlassen wird. Erreicht der Zähler einen bestimmten Wert, so wird unser Feuerwerk aktiv, aber jetzt haben wir schon wieder vorgegriffen. Die Semaphore, die unsere Tür (siehe Beispiel oben) darstellt, die wir auf (0) und zu (1) machen können, setzen wir auf 0, d.h. daß unsere Interrupt-Routine momentan offen und damit nicht benutzt ist. Nun holen wir uns aus der (privilegierten) Adresse $70 die Adresse der aktuellen VBL-Routine, sichern diese und tragen unsere eigene (*vbl_count*) ein, so daß ab dem nächsten VBL unsere Routine angesprungen wird - ab jetzt ist das Feuerwerk initialisiert, d.h. daß ab diesem Moment eine gewisse Anzahl von VBLs abgewartet und dann das Feuerwerk gestartet wird (siehe unten). Zum Schluß der *ikbd_on*-Routine verbiegen wir noch die Einsprungadresse des Line-A. Stellen Sie sich vor, das Betriebssystem sei gerade mit Line-A dabei ein paar Linien auf dem Bildschirm zu zeichnen und auf einmal kommt ihm ein Feuerwerk dazwischen,

```
          {
             ch_dist=font->max_cell_width;
             x_start=res_x/2-len*ch_dist/2;
          }
          y=res_y/2+(text_nr-3)*font->form_height*2;
             i=0;

          /* Buchstaben des Strings ausgeben */
          while(intro_str[text_nr][i] && *(long*)vbl_cntr)
          {
             x=x_start+i*ch_dist;
             if (intro_str[text_nr][i]!=' ') /* Bei ' ' keine Rakete */
             {
               curve((float)res_x,absf(res_x-x),1.0,(float)(res_y-y),
                     -1.0,res_x,res_y,15);
               a008(intro_str[text_nr][i],x,y,font);
             }
             i++;    /* nächstes Zeichen */
          }
      }
}


/**********************************************************/
/* firework ist für das eigentliche Feuerwerk zuständig.  */
/* Wegen der Zugriffe auf beispielsweise Hardwareadressen */
/* läuft die Routine nur im Supervisor-Modus...           */
/**********************************************************/

void firework(res)
int res;
{
   float startx, breite, breite1, weite, hoehe;
   int res_x, res_y;
   float direction;
   int t, num=0;
   int mp[EXPL_MAX][4], scratch_buf[16];
   int linie=0xffff;
   long routines, *fonts;
   int *screen_poi;
   int coll;
   LINE_A la_dup;


   res_x=639;                /* Auflösung wählen */
   res_y=399;
   switch(res)
    {
         case 0:
         res_x=319;
         case 1:
         res_y=199;
   }

    coll=*(int*)0xff8240;     /* Hardwareregister der Farbpalette */
    *(int*)0xff8240=0;        /* Farbe 1 auf Schwarz setzen*/

   linea=a000(&fonts, &routines);      /* Line-A-Bereichsadresse
                                            ermitteln */
   set_line_a(&linie,0,1,0,0,0,0,0);/* Line-A-Parameter setzen */
   linea->WMODE=MD_REPLACE;          /* Zeichenmodus */
   a005(0,0,res_x,res_y,0);          /* Bildschirm schwarz füllen */
   set_line_font((FONT_HDR*)*(fonts+1),1,0,0,0,0,0,(int*)
                  scratch_buf);
   intro(res_x, res_y, *(fonts+1));

   while(*(long*)vbl_cntr)          /* hat sich Tastatur gemeldet ? */
   {                                    /* Nein */
      startx=zufall()/1023.0*res_x;     /* Startposition ermitteln */
      breite= startx;                   /* Abstand zum linken Rand */
      breite1=(res_y-startx);           /* Abstand zum rechten Rand */
      direction=-1;                     /* Richtung nach links */
      if (breite1>breite)               /* Rechter Abstand gröPer ? */
      {                                 /* Ja */
         direction=1;                   /* Richtung nach rechts */
         breite=breite1;                /* Breite = rechte Breite */
      }

         /* Weite zufällig aufgrund maximaler Weite ermitteln
            Höhe zufällig aufgrund Auflösung ermitteln */

    weite=zufall()/1023.0*(breite-AUSBREITUNG/2-MIN_WEITE)+MIN_WEITE;
    hoehe=zufall()/1023.0*(res_y-MIN_HOEHE-AUSBREITUNG/2)+MIN_HOEHE;

      weite/=1.3;    /* ein bischen weniger */

```

→

welches auch mit Line-A gezeichnet werden soll - dies geht eventuell ziemlich schief. Deshalb verbiegen wir uns den Einsprung ins Line-A auf eine eigene Routine, die sich merkt, daß momentan Line-A am arbeiten ist und das Line-A-Arbeitsflag erst dann wieder löscht, wenn die Arbeit im Line-A beendet ist. Da das Verbiegen des Line-A-Handlers aber recht kompliziert ist, soll darauf gleich nochmal detailliert eingegangen werden.*ikbd_off* macht im Prinzip genau das gleiche wie *ikbd_on*, nur umgekehrt, d.h. daß es die ganzen verbogenen Zeiger wieder restauriert also mit seinen Originaladressen belegt.

Achtung Umleitung

Eine der trickreichsten Routinen im Feuerwerk ist das Verbiegen des Line-A-Vektors. Um zu verstehen, warum dies so schwierig ist, muß man sich erst einmal anschauen, was der Line-A-Handler so alles treibt. Dazu schauen Sie sich bitte Listing 2 an. Wenn Sie sich mit dem 68000 schon ein wenig beschäftigt haben, so wissen Sie, daß der Prozessor bei Unterprogrammsprüngen die Rücksprungadresse auf den Stack legt. Bei der Verarbeitung von Ausnahmevektoren, wie es ein Line-A-Befehl ist, werden Statusregister und die Adresse, an der der Befehl stand, der die Ausnahme bewirkte, auf den Stack gelegt. Dann wird bei einem Line-A-Maschinencode der Sprungvektor aus Adresse $28 geholt und zu der in Listing 2 zu findenden Routine verzweigt. Wie bekommt aber der Line-A-Handler heraus, ob der Befehl A000, A001 usw. geheißen hat? Ganz einfach und gemein: auf dem Stack steht doch die Adresse, wo die Ausnahmeverarbeitung aufgetreten ist, d.h. also die Adresse, an der der Line-A-Befehl steht. Der Line-A-Handler schaut also tatsächlich im Programm nach, welcher Befehl dort steht. Dann blendet er die oberen acht Bit des Zwei-Byte-Line-A-Befehls aus und benutzt ihn, mit vier multipliziert, als Index auf eine Sprungtabelle, in der die Adressen der Line-A-Routinen stehen.

Das gemeine ist folgendes: Wir wollen ja eine Routine schreiben, die ein Flag setzt, bevor die Line-A-Routine ausgeführt wird und dieses löscht, wenn die Line-A-Routine ausgeführt worden ist. Das bedeutet, wir wollen Line-A aufrufen und es soll zu uns zurückkehren. Dies macht man im allgemeinen durch einen JSR-Befehl, der die Rücksprungadresse auf den Stack legt, allerdings geht dies hier nicht so

```
        /* Koordinaten der ersten Explosion, Y ergibt sich aufgrund des
           letzten Punktes der Kurve, der zurückgeliefert wird */
        mp[0][0]=(int)(startx+1.3*weite*direction);  /* X, Y */
        mp[0][1]=(int)curve(startx, weite, 1.3, hoehe, direction,
                            res_x, res_y, 30);
         mp[0][2]=STERNBREITE;                           /* 'Zeit' */
         mp[0][3]=1;                                     /* Farbe */


            /* Ermitteln der Koordinaten der Sterne */
         for(t=1; t<N_EXPL && *(long*)vbl_cntr; t++)
         {
           if(zufall()<768) /* Diesen Stern gibts nur dann */
           {
              /* x: */
               mp[t][0]=mp[0][0]+(int)(zufall()/1023.0*AUSBREITUNG
                                        -AUSBREITUNG/2);
              /* y: */
               mp[t][1]=mp[0][1]+(int)(zufall()/1023.0*AUSBREITUNG
                                        -AUSBREITUNG/2);
              /*Zeitabstand: */
               mp[t][2]=STERNBREITE+(zufall()/
                                        1023.0*N_EXPL*0.5*STERNBREITE);

               mp[t][3]=(int)(zufall()/1023.0*16);            /* Farbe */
           }
           else
               mp[t][2]=0;             /* Dieser Stern geht nicht auf */
         }

         do_stars(mp, N_EXPL);         /* Gesamtexplosion durchführen */

         if (*(long*)vbl_cntr)      /* Falls keine Aktion am Rechner */
           pause((int)(zufall()*3));/* zufällige Wartezeit einleiten */

         if (++num==1)                    /* nach erster Explosion */
         {
           linea->WMODE=MD_REPLACE;        /* Intro-Bild */
           a005(0,0,res_x,res_y,0);        /* löschen */
         }
       }
       *(int*)0xff8240=col1;            /* Farbe0 wieder zurücksetzen */
}

/*********************************************************/
/* Die folgende Routine do_stars() führt die (teilweise  */
/* ineinandergehenden) Explosionen durch.                */
/*********************************************************/
do_stars(star, number)/*Star enthält Sternkoordinaten und Zeitoffset
*/
int star[][4], number;/* number: Anzahl der Sterne */
{
     int flag,t;       /* Flag!=0 bedeutet: noch nicht alle Sterne */
                                /*                    gezeichnet */

     flag=number;          /* Flag auf Anzahl der Sterne setzen */

       while(*(long*)vbl_cntr  && flag) /* Bis alle Stern fertig */
       {
         for(t=0; t<number; t++)    /* Ein Schritt aller Sterne
                                       zeichnen */
            stern(star[t][0],star[t][1],star[t][2],star[t][3]);

         pause(25);                     /* Einen Moment warten */
         flag=number;                   /* Flag auf Gesamtanzahl setzen */
         for(t=0; t<number; t++)        /* Diesen Schritt aller Sterne */
         {
           stern(star[t][0],star[t][1],star[t][2],0);/*wieder löschen*/
             star[t][2]--;       /* Sternzeit erniedrigen */
             if (star[t][2]<=0) /* Sternzeit dieses Sterns abgelaufen*/
               flag--;                   /* Ein Stern mehr fertig */
         }
       }
}

/* Die nächste Routine zeichnet eine Kurve eine Parabel aufgrund der
   Angabe des Startpunktes und des Hochpunktes, sowie der Richtung
   des 'Fluges'. Faktor=1 bedeutet Flug bis zum Hochpunkt. GröPer 1
   ergibt ein Fliegen darüberhinaus. In diesem Fall wird der
   Endpunkt interessant und daher zurückgegeben. */

float curve(startx, weite, faktor, hoehe, direction, res_x, res_y,
            time)
float startx, weite, faktor, hoehe, direction;
int res_x, res_y, time;
{
```
→

einfach, da Line-A mit einem RTE abschließt, was andererseits dadurch zu umgehen wäre, daß man vor dem JSR das Statusregister auf den Stack schreibt. Die unangenehme Tatsache bei Line-A ist aber, daß es den Line-A-Code, wie oben beschrieben, aufgrund der Rücksprungadresse aus dem Programmcode holen möchte. Wenn aber eine Zwischenroutine (wie die unsere) Line-A mit einem JSR aufruft, versucht Line-A nach wie vor den Befehl aus dem Programmcode zu holen. Allerdings geht dies diesmal schief, den Line-A greift wegen der Rücksprungadresse des JSR auf den Code unserer Zwischenroutine zu, der normalerweise alles andere als einen Line-A-Befehl enthält. Wir müssen uns also etwas einfallen lassen und mit den gleichen Waffen wie Line-A zurückschlagen - die Lösung finden Sie in *line_trap*.

Zunächst machen wir das gleiche wie Line-A und holen uns den Line-A-Befehl aus dem Programm und errechnen uns die Adresse des Rücksprungs. Jetzt merken wir uns in *line_flag*, daß wir gleich in den Line-A-Handler springen. Den Befehl schreiben wir an die Stelle des Codes, deren Adresse wir auf den Stack schreiben. Das gaukelt Line-A das richtige Programm vor. Es greift auf diese Adresse zu, um den Line-A-Code zu holen, und springt später dahinter, wo nach getaner Arbeit das *line_flag* wieder gelöscht und die Ausnahmeverarbeitung (exception) verlassen wird. Und schon haben wir eine Zwischenroutine, die ein Flag gesetzt hat, solange das Betriebssystem sich in Line-A befindet....

VBL very busy lines

Nach soviel Vorarbeit kommen wir nun zur eigentlichen VBL-Routine, die wir Schritt für Schritt durchgehen werden. Zunächst retten wir mal ein paar Register, die zum Schluß natürlich wieder restauriert werden. Immer wenn in unsere VBL-Routine gesprungen wird, sorgt der Befehl *clr.b vbl_flag* dafür, daß *vbl_flag* gelöscht wird. Dies benutzen wir weiter unten an der Stelle *wait_for_vbl*, wo wir warten bis ein VBL vergangen ist. Wieso warten wir auf den VBL, obwohl wir gerade drin sind? Von dieser Denkweise müssen Sie sich einfach freimachen: Auch wenn wir gerade in einer VBL-Routine drin sind, wird kurze Zeit später die gleiche Routine wieder angesprungen. Wie Sie aber von oben noch wissen, sorgen wir kurz nachdem wir in unserer Routine sind dafür, daß die folgende

```
    float rx, x, iy, lx=0, ly=0;

   for (x=0; x<weite*faktor && *(long*)vbl_cntr; x+=weite/30.0)
   {
     iy=(-hoehe/weite/weite*x*x+2*hoehe/weite*x);/* momentane Höhe */
     rx=x*direction;                               /* Richtung */
                                                /* Strich zeichnen */
     a003((int)(startx+lx),(int)(res_y-ly),(int)(startx+rx),
          (int)(res_y-iy),(int)(Random()/1023.0*16)|1);
     pause(time);                               /* warten */
                                                /* Strich löschen */
     a003((int)(startx+lx),(int)(res_y-ly),(int)(startx+rx),
          (int)(res_y-iy),0);
     lx=rx;                                    /* aktueller Endpunkt wird */
     ly=iy;                                    /* Anfangspunkt neuer Linie */
   }
    return(res_y-iy);                                /* letzte Höhe */
 }


 stern(mx,my,i,c)/*Einen Stern zeichnen,
                   mit Mittelpunkt, Lebenszeit*/
 int mx,my;         /* und Farbe */
 int i,c;
 {
    int mxpl, mxml, mxm, mxp;
    int mypl, myml, mym, myp;

    if (i<=0 || i>STERNBREITE) /* Zeit <0: Stern schon 'tot'; >Stern
                                 breite */
       return;                /* bedeutet Stern noch nicht geboren */

    i=STERNBREITE-i;    /* umdrehen */

    mxpl=mx+ST_L+i/2;   /* Abstand in alle 4 Richtungen ermitteln */
    mxml=mx-ST_L-i/2;
    mxm=mx-i;
    mxp=mx+i;
    mypl=my+ST_L+i/2;
    myml=my-ST_L-i/2;
    mym=my-i;
    myp=my+i;

    a003(mx-ST_L-i,my,mxm,my,c);  /* 8 Linien ziehen */
    a003(mx+ST_L+i,my,mxp,my,c);
    a003(mx,my-ST_L-i,mx,mym,c);
    a003(mx,my+ST_L+i,mx,myp,c);
    a003(mxml,myml,mxm,mym,c);
    a003(mxpl,myml,mxp,mym,c);
    a003(mxpl,mypl,mxp,myp,c);
    a003(mxml,mypl,mxm,myp,c);
 }

 void pause(aktuell)                      /* Ein paar '200Hz' warten */
 int aktuell;
 {
     long a;

     a=*(long*)0x4ba;                     /* 200Hz-Zähler */

   /* Warten bis Differenz des aktuellen Zählerstandes zu Anfangswert
            gröPer gleich gefordertem ist, oder Taste berührt */
   while( ((*(long*)0x4ba)-a)<(long)(aktuell/4) && *(long*)vbl_cntr);

 }

 long zufall()                                /* 'Zufallszahl' errechnen */
 {
    random= seed%1023;                        /* 0 - 1023 */
    seed  = (25173*seed+13849)&65535;  /* Berechnungsformel */
    return(random);                           /* zurückgeben */

 }

 /* Die folgende Routine dient dem Setzen von */
 /* diversen LINE-A-Variablen.                */
 set_line_a(pattern, pat_nr, plane_flag, plane_no,
              clip_flag, cl_x1, cl_y1, cl_x2, cl_y2)
 int *pattern, pat_nr;                     /* Muster */
 int plane_no;                             /* Anzahl der Planes */
 int plane_flag, clip_flag;                /* Farbflag, Clip-Flag */
 int cl_x1, cl_y1, cl_x2, cl_y2;          /* Clip-Bereich */
 {
    linea->PATPTR  = pattern;         /* Muster-Adresse */
    linea->PATMSK  = pat_nr;          /* Musterhöhe */

```

→

VBL-Routine um uns herumläuft, was durch die Abfrage *tst.b semaphore* geschieht. Nachdem die Abfrage durchgeführt und beim ersten Mal noch nicht erfüllt ist, kann die Bearbeitung der folgenden Zeilen erfolgen, wo wir als erstes die Semaphore setzen, das heißt alle folgenden VBL-Aufrufe kommen nur bis zur Abfrage, werden aber diesmal direkt nach *vbl_end1* und damit um unserer Routine herumspringen. Weitere Kriterien dafür, daß das Feuerwerk vorerst nicht zur Geltung kommt, ist die Tatsache, daß das Betriebssystem sich momentan in Line-A befindet (*line_flag* gesetzt! - siehe oben) oder es wird momentan eine Diskettenoperation durchgeführt, was man in der Adresse $43e (flock) nachschauen kann.

Komfort

Was wäre ein Bildschirmschoner mit Feuerwerkausführung, wenn er nicht auch etwas an Komfort bieten würde. Führt man die Maus in die linke obere Ecke, so bemerkt dies die VBL-Routine und löscht immer wieder den internen Zähler. Dies hat zur Folge, daß er nie den Endwert erreicht und das Starten des Feuerwerks unterdrückt wird (*vbl_supress*), beispielsweise wenn man am Ausdrucken längerer Texte ist. Möchte man das Feuerwerk sofort starten, ohne daß der Rechner eine gewisse Zeit wartet, so führt man die Maus in die linke untere Ecke, drückt dann beide Maustasten gleichzeitig, läßt die Maustasten wieder los und die Maus in dieser Ecke stehen. Diese Abfrage ist natürlich etwas komplizierter, da man sich hier merken muß, ob bei gleichzeitigem Drücken der Maustasten die Maus in der linken unteren Ecke ist, wobei danach ein Flag (*fireflag*) gesetzt werden muß, denn es ist sicherlich nicht sinnvoll, daß der Benutzer andauernd die Maustasten gedrückt halten muß... Das Beginnen des Feuerwerks erzwingen wir übrigens dadurch, daß wir den Zähler auf den Endwert setzen (*vbl_cntr=wait_time*). Die Mauskoordinaten holen wir uns aus den in den letzten Line-A-Folgen schon oft angesprochenen negativen Line-A-Variablen, die, es sei nochmals erwähnt, einen völlig legalen Zugriff auf die Mauskoordinaten erlauben, ohne daß eine Betriebssystemfunktion aufgerufen werden muß.

Seitenwechsel

Nach diesen Mausabfragen geht es an der Stelle *vbl_cnt/2* weiter. Sind wir einmal

```
   linea->MFILL = plane_flag;      /* alle Farbbits benutzen ? */

   if (plane_no>0)                 /* 0 nicht erlaubt !!!!! */
   {
    linea->PLANES= plane_no;
    linea->WIDTH = (plane_no>1) ? 160:80;/* Bytebreite des Bildes */
   }
   linea->CLIP    = clip_flag;     /* Clipping ein/ausschalten */
   linea->XMINCL  = cl_x1;         /* Bereich setzen */
   linea->YMINCL = cl_y1;
   linea->XMAXCL = cl_x2;
   linea->YMAXCL = cl_y2;
}

/************************************************************/
/*                                                          */
/*      Setzen der Farbe in unterschiedliche Variablen      */
/*                                                          */
/************************************************************/

void set_line_color(color) /* Farbe in Line-A-Variablen schreiben */
int color;
{
   linea->COLBIT_0 = color&1;           /* Bit 0 der Farbe */
   linea->COLBIT_1 = (color&2)==2;      /* Bit 1 der Farbe */
   linea->COLBIT_2 = (color&4)==4;      /* Bit 2 der Farbe */
   linea->COLBIT_3 = (color&8)==8;      /* Bit 3 der Farbe */
}

void set_line_font(font,col_fg,col_bg,direc,change,style,rot,
                                                      scratch)
int col_fg, col_bg, direc,change,style,rot;
int *scratch;
FONT_HDR *font;
{
   int max;

   if (!font)
      return;

   linea->XDDA=    0x8000;            /* das ist immer so */
   linea->SCALE=   (change>0);        /* Wenn folgender Wert>0 */
   linea->DDAINC= change;             /* Änderung der Skalierung */
   linea->SCALDIR= direc;             /* VergröPern=1/Verkleinern=0*/
   linea->MONO=(font->flags&8>0);     /* gleiche Breiten ? */
   linea->STYLE=    style;            /* Texteffekte */
   linea->LITEMASK= font->lighten;    /* Maske für Hellschrift */
   linea->SKEWMASK= font->skew;       /* Maske für Kursivschrift */
   linea->WEIGHT=   font->thicken;    /* Fettschriftänderung */
   linea->LOFF= font->left_offset;    /* linker Offset für
                                         Kursivschrift */
   linea->ROFF= font->right_offset; /* rechter Offset für Kursiv
                                   schrift */
   linea->CHUP= rot;               /* Drehrichtung: 0,900,1800,2700 */
   linea->TEXTFG=  col_fg;           /* Textfarbe */
   linea->TEXTBG=  col_bg;            /* Hintergrund */
   linea->SCRTCHP= (long)scratch;     /* Puffer für Texteffekte */
   max= ((font->max_cell_width)>(font->form_height)) ?
         font->max_cell_width:font->form_height;
   linea->SCRPT2=   max;         /* maximale GröPe eines Zeichens */
   linea->DELY =   font->form_height;
   linea->FBASE=   font->dat_table;
   linea->FWIDTH=  font->form_width;
}


LINE_A *a000(font_poi, routine_poi) /* Gibt Zeiger auf LINE_A-Struk-
                                       tur nzurück */

long *font_poi, *routine_poi;
{
   asm{
      dc.w 0xa000               /* Line-A-Initialisierung aufrufen */
      move.l font_poi(A6), A0  /* Zeiger auf die Zeichensätze */
      move.l A1, (A0)            /* abspeichern */
      move.l routine_poi(A6), A0/* Zeiger auf die Line-A-Routinen */
      move.l A2, (A0)            /* abspeichern */

      }
}                           /* Zeiger auf LINE-A-Struktur in D0 */

/************************************************************/
/*       Zum Setzen eines Punktes wird A001 benutzt         */
/*                                                          */
/*                          A001                            */
/*                                                          */
/************************************************************/
```

→

Lohn & Gehalt ST

Die Lohn- und Gehaltsabrechnung für mittlere Betriebe

Leistungsmerkmale

- Einfach zu bedienende, praxiserprobte Abrechnung
- Brutto-, Nettolohnabrechnung, Pauschalierung
- Unbegrenzte Mitarbeiteranzahl
- Keine Lohnarten- Einschränkung
- Individuelle Formulare
- Lohn- und Gehaltsabrechnung, Lohnjournal, Lohnkonto, Krankenkassenliste, Zahlungsaufträge etc.
- Aufteilung auf Kostenstellen möglich
- Durchschnittslohn
- Einmalzahlungen, auch unter Berücksichtigung der Märzklausel
- Berlinfähige Abrechnung
- Einstieg während des laufenden Geschäftsjahres möglich

DM COMPUTER

7530 Pforzheim • Kaiser-Friedrich-Straße 8 • Telefon 0 72 31/2 60 91

Nikolaistraße 2
8000 München 40
West-Germany

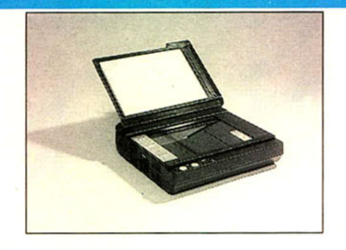
PRINT TECHNIK

Tel. 0049-89-368197
FAX: 0049-89-399770

Wir stellen aus: Systems München,

Schweiz: Microtron 0041-32-872429 NL 010-4507696 VISA/Eurocard Accepted

Austria: Print-Technik-Wien 0043-222-5973423 VISA/Eurocard Accepted



Neuer Superpreis: 1198.-

inkl. OCR-Schrifterkennung
Univ. Scanner, Drucker, Kopierer

Dieses mit 200 DPI arbeitende Bilderfassungsgerät ist die ideale Arbeitshilfe für alle Anwender, die über Geräte mit einem Mega-Speicher verfügen (1040, ein Mega, oder aufgerüstete Einheiten). Durch rationellste Produktionsmethoden und günstigen Einkauf des Thermo-Kopierers ist uns nochmals eine Preissenkung für dieses Gerät gelungen. Der fertig-anschließbare Universal-Scanner ist zur Zeit für DM 1198,- erhältlich.
Ein absoluter Preishit für jeden ATARI-Nutzer.

Professional Scanner 2998.-

mit OCR-Junior inkl. Ganzseitenmalprogramm ROGER PAINT OCR Junior selbstlernende Schrifterkennung PEGASUS + ST 1 Raster Vektor Konvertierungsprogramm
300 x 300, 300 x 600, 600 x 600 DPI-Auflösung und 64 Graustufen, einschl. Zeichenprogramm und OCR-Schrifterkennung.
Diese Scannerneuheit für den Industrie- und DTP-Bereich stellt einen absoluten Preishit dar. Mit ihm lassen sich sowohl Halbton als auch binäre Vorlagen scannen und ablegen und mit allen auf dem Markt befindlichen Programmen (auch Calamus) weiterverarbeiten.
Das mitgelieferte Schrifterkennungsprogramm erlaubt das Umsetzen von Text in ASCII-Zeichensatz und ist durch seine Lernfähigkeit von hoher Effektivität.

Videodigitizer PRO 8900 für ATARI
Der Videodigitizer PRO 8805 liefert die höchste Auflösung, die bei Verwendung einer normalen Videokamera möglich ist: 1024 Punkte in 512 Zeilen. Gleichzeitig digitalisiert er mit einer Genauigkeit von 7 Bit, was einer Anzahl von 128 Graustufen entspricht. Technische Daten des PRO 8805: Bildformate: Neochrome, IMG, Doodle, Spat. Ausdruck auf: NEC P6/P7, ATARI Laser. Auflösung: 320 x 200, 640 x 200, 640 x 400, 512 x 512, 1024 x 512. Graustufen: 128 (7 Bit). Anschluß: ROM-Port des ATARI ST. Eingangssignal: BAS oder FBAS. S/W und Farbmonitor.
Preis: DM 498,-

Neue Colorsoft von Imagic
16 Farben aus 4096/Zusatzsoft zum PRO 8900 **Preis: DM 98,-**

PRO 8900 mit RGB-Filter + Imagic Soft.
Der »Farb«-Digitizer **DM 748,-**

Realtizer für ATARI ST
Der REALTIZER ist ein in den ROM-Port einsteckbares Modul zur rasanten Digitalisierung von Videobändern aller Art. Die Auflösung beträgt 320 x 200 Punkte, wobei der Farb- und Monochrome-Modus (640 x 400) des ATARI ST unterstützt wird. Die Auflösung: 16 Graustufen. Pro Graustufe beträgt die Digitalisierungszeit 1/25 Sekunde.
Automatische Helligkeits- und Kontrastregelung. **Preis: DM 198,-**

OCR-Junior Schrifterkennung
Selbstlernende Schrifterkennung zu Universalscanner für ATARI ST.
Preis: DM 198,-

Romportstecker
Freier Druckerport beim Universalscanner. Ermöglicht Sofortausdruck z.B. mit NEC P6/P7. **Preis: DM 198,-**

RGB Splitter
Der RGB-Splitter zerlegt jedes Farb-Videosignal in seine Grundfarben Rot, Grün und Blau. Mittels Drehschalter kann jede Grundfarbe und Schwarz/Weiß an einen Videoausgang geschaltet werden. Passend für alle Videodigitizer mit Farbdigitalisierungssoftware (z. B. PRO 8805).
Noch nie erreichte Farbbildqualität. **Preis: DM 298,-**

Videotext Dekoder
Zum Anschluß an den ROM-Port. Kann mit jedem Videosignal betrieben werden. Läuft auf Farb- oder S/W-Monitor. Seitenweises Aufrufen – Automatisches Blättern – Seiten halten – Speichern und Laden der empfangenen Seiten im Text- oder Bildformat – Textausdruckmöglichkeiten über beliebige Drucker. **Preis: DM 298,-**

PRODUCTION – IMPORT – EXPORT – DISTRIBUTION

bis hierher gekommen, erkennen wir, daß an dieser Stelle der schon oft erwähnte Zähler *vbl_cntr* inkrementiert wird. Diesen vergleichen wir mit dem Wert, den wir in *wait_time* finden und der im Hauptprogramm auf einen bestimmten Wert gesetzt wurde (300 oder 9000). Ist der Endwert noch nicht erreicht, springen wir, wie bereits bekannt, nach *vbl_end*. Ist der Zähler aber an dem eingestellten Endwert angekommen, beginnt das eigentliche Feuerwerk. Dazu schalten wir auf einen anderen Bildspeicherbereich (Bildschirmseite) um, was mit der Hardwareadresse $FF8201 (High-Byte) und FF8203 (Mid-Byte) vonstatten geht, die die Anfangsadresse des Bildspeichers enthalten - ein Low-Byte gibt es nicht, d.h. ein Bildspeicher muß immer an einer Adresse beginnen, deren Low-Byte 0 ist! Eine elegante Methode diese ein Byte auseinanderliegenden Adressen zu lesen oder beschreiben, bietet der MOVEP-Befehl. Leider ist er im Megamax-Laser-C-Inline-Assembler (Donaudampfschiff...) nicht enthalten, so daß ich ihn einfach als HEX-Code hineingeschrieben habe! Zunächst merken wir uns die alte physikalische Bildschirmadresse. Danach setzen wir endlich den Interruptlevel herunter. Können Sie sich noch erinnern? Nur dann, wenn ein Interrupt niedrigerer Priorität ankommt, wird dieser akzeptiert und damit weiter VBLs durchkommen, erniedrigen wir den aktuellen VBL-Level auf 3 (der VBL hat eine Priorität von 4). Nun schalten wir auf unseren anderen Bildschirm, um das aktuelle Bild nicht zu zerstören, in dem wir die Adresse unseres Speicherbereiches (*screen_poi*) in die Hardwareadressen des Video-Chips (siehe oben) schreiben.

Jetzt kommts: Wichtig ist, daß wir retten, was zu retten ist, bevor wir in die C-Routine (!) springen, die das eigentliche Feuerwerk übernimmt. Dazu werfen wir alle Register auf den Stack. Bevor wir aber das Feuerwerk aufrufen, wird noch die aktuelle bei der Initialisierung ermittelte Auflösung auf den Stack als Parameter für *firework()* geschrieben. Nachdem das Feuerwerk zurückkommt (eine Beschreibung folgt unten), reparieren wir den Stack und restaurieren alle Register. Da der Video-Chip es nicht so gerne hat, wenn man mitten im VBL seine Bildschirmadresse umschaltet, sprich es kann zum Absturz des Rechner führen, warten wir bis ein VBL vergangen ist. Dies machen wir mit dem oben schon angedeuteten *vbl_flag*, das wir an dieser Stelle zunächst einmal setzen. Nun warten wir,

```
void a001(x,y, color)               /* Punktkoordinate und Farbe */
int x,y,color;
{
   linea->INTIN[0]=color;           /* Farbe */
   linea->PTSIN[0]=x;               /* Koordinate X */
   linea->PTSIN[1]=y;               /* Koordinate Y */
   asm{
      dc.w 0xa001                   /* Line-A-Put_Pixel */
      }
}

/***********************************************************/
/*      Zum Zeichnen einer Linie wird A003 benutzt         */
/*                                                         */
/*                        A003                             */
/*                                                         */
/***********************************************************/

int a003(x1,y1,x2,y2,color)          /* Linie ziehen */
int x1,y1,x2,y2,color;
{
   /* WMODE und LNMASK wird extern gesetzt */
   linea->X1=x1;                    /* Koordinaten setzen */
   linea->Y1=y1;
   linea->X2=x2;
   linea->Y2=y2;

   set_line_color(color);           /* Farbe aufteilen */

   linea->LSTLIN  = 1;              /* letzten Punkt zeichnen */

   asm{
      dc.w 0xa003                   /* Linie ziehen */
      }
}                                   /* geshiftetes Ergebnis */

/***********************************************************/
/*           Zeichnen eines ausgefüllten Rechtecks         */
/*                                                         */
/*                        A005                             */
/*                                                         */
/***********************************************************/

int a005(x1,y1,breite,hoehe,color)
int x1,y1,breite,hoehe,color;
{
   int tausch;

   if (linea->PATPTR == 0)           /* kein Muster gesetzt */
      return;

   if (!breite || !hoehe)        /* Fläche hat keine Breite/Höhe */
      return;

   linea->MFILL=1;                  /* alle Farbebenen */
   linea->X1=x1;                    /* Koordinaten setzen */
   linea->Y1=y1;
   linea->X2=x1+breite;
   linea->Y2=y1+hoehe;

   set_line_color(color);

   asm{
      dc.w 0xa005
      }
}

/***********************************************************/
/*                 Ausgabe eines Zeichens                  */
/*                                                         */
/*                        A008                             */
/*                                                         */
/***********************************************************/

int a008(ch, x, y, font)
unsigned char ch;
FONT_HDR *font;
{
   int back_x;

   if (!font)
      return;


   if (font->flags&8)
```

→

bis dieses Flag Null wird, was dann geschieht, wenn irgendwann ein VBL-Interrupt ausgelöst in unsere Routine oben hineinspringt und dann dort den Befehl *clr.b vbl_flag* ausführt. Ab dann darf auch unsere Routine wieder weiterlaufen, die sogleich das Statusregister wieder restauriert und die Bildschirmadresse wieder auf das Originalbild setzt. Wie oben manipulieren wir natürlich auch die logische Bildschirmadresse, damit Line-A auch wieder auf den Originalbildschirmbereich zugreift und nicht mehr auf unser Feuerwerkbildschirm. Haben wir dies getan können wir unsere Tür wieder aufmachen, sprich die Semaphore entlasten (löschen) und den Zähler auf den Anfangswert Null setzen. Was nun kommt ist Routine: Register restaurieren und durch den Original-VBL-Vektor springen, denn die nachfolgenden Routinen wollen ja auch bedient werden.

```
    {
     linea->SOURCEX= font->off_table[ch-(font->first_ade)];
     linea->DELX =    font->off_table[ch+1-(font->first_ade)]-(linea-
                                                         >SOURCEX);
    }
    else
    {
       linea->SOURCEX=(ch-(font->first_ade))*(font->max_cell_width);
       linea->DELX =  font->max_cell_width;
    }
    linea->SOURCEY= 0;
    linea->DESTX=   x;
    linea->DESTY=   y;

    back_x=x+linea->DELX
           +((linea->STYLE&4)!=0)*(font->left_offset+font-
                                   >right_offset)
           +((linea->STYLE&16)!=0)*2;
    asm
    {
       dc.w 0xa008
    }
    return(back_x);
 }
```

```
*/ Listing 2: Der Line-A-Handler */

Die Adresse der folgenden Routine ist in Adresse $28-$31 zu finden:*/

 LINE_A_HANDLER:
 MOVE.L  2(A7), A1          */ Fehleradresse auf dem Stack
 MOVE.W  (A1),D2            */ Code an der Fehleradresse
 AND.W   #$FFF, D2          */ untere Bits ausblenden
 ADDQ.L  #2, A1             */ A1 auf Adresse nach Line-A-Befehl setzen
         MOVE.L  A1, 2(A7)  */ Adresse auf Stack als Rücksprungadresse
         CMP.W   #$F, D2    */ Code<=15
         BHI      ENDE      */ Nein. Code >15 nicht erlaubt -> Ende
         LSL.W   #2, D2     */ mit vier multiplizieren, als Index in
         MOVE.L  (ROUT_TAB,D2), A1  */ eine Line-A-Routinen-
                                       Adreßtabelle verwenden
         MOVEM.L D3-D7/A3-A5, -(A7) */ Register retten
         JSR       (A1)             */ Line-A-Routine aufrufen
         MOVEM.L (A7)+, D3-D7/A3-A5 */ Register restaurieren
 ENDE:
         RTE                        */ Rücksprung von der Exception
```

Here comes...

... the show. Zum Schluß möchte ich noch ein wenig zu dem eigentlichen Feuerwerk sagen, was aber im Gegensatz zu den eben besprochenen Maschinenroutinen recht einfach zu verstehen ist. Im Prinzip besteht es aus einer Einleitung, bei der ein Text schön zentriert auf den Bildschirm geschrieben wird. Dabei erscheint jeder Buchstabe von einer kleinen Explosion begleitet. Danach verschwindet dieser Text, sonst hätten wir wohl kaum einen Bildschirmschoner, sondern eher einen Texteinbrenner, und das eigentliche Feuerwerk beginnt. *firework()* benutzt dabei als wichtigste Routine die Funktion *curve()*, die eine Parabelkurve mit vielen veränderbaren Parametern wie Höhe, Startpunkt, Richtung, Geschwindigkeit und so weiter enthält. Die Funktion *Stars()* ist mit *firework()* zusammen in der Lage, viele ineinandergehende Sterne zu zeichnen. Interessant ist auch die Funktion *Pause()*: Da es uns nicht möglich ist, aus dem Interrupt heraus eine Betriebssystemfunktion aufzurufen, wurde die Zeitmessung auf den internen 200 Hz-Timer zurückgeführt, dessen aktuellen Wert man in Adresse $4ba finden kann.

Das gleiche Problem hat man mit der Generierung einer Zufallszahl, da uns die Routine aus dem Betriebssystem verschlossen bleibt. Hier hilft nur ein eigens programmierter Algorithmus, der aufgrund einer bekannten Formel eine Zahlenfolge erzeugt, die möglichst abwechselnd ist. Natürlich sind die Zahlen nicht wirklich zufällig, sondern nur recht gut verteilt. Allerdings holen wir uns den Anfangswert aus der Betriebssystemroutine *Random()* - wir hätten aber auch zum Beispiel den 200 Hz-Timer als Anfangswert verwenden können.

Das Ende vom Lied

Lange hat unser Streifzug durch die Gefilden des Line-A gedauert und dabei habe ich und hoffentlich auch Sie viel darüber gelernt. Ich hoffe, daß ich mit dieser Reihe endlich mal ein einigermaßen vollständiges Nachschlagewerk in Sachen Line-A schaffen konnte. Die ersten Rückmeldung bezüglich Anwendung, Fragen und Anmerkungen habe ich schon bekommen, so daß man irgendwann mal einige Tips, Tricks und bekannte Fehler zusammentragen könnte. Zum Abschluß möchte ich noch einmal darauf hinweisen, daß Sie sich immer überlegen sollten, ob Sie nicht besser doch GEM-Funktionen als die des Line-A verwenden. Viele von uns haben inzwischen den ATARI TT auf der Düsseldorfer Messe gesehen (OK, über das Design läßt sich wieder einmal streiten) und im Zuge der Kompatibilität sollte man schließlich auf oberster Ebene des Betriebssystems programmieren, damit einfach garantiert bleibt, daß die Programme später auch auf anderen Betriebssystemversionen laufen.

P.S.: Lassen Sie das Feuerwerk mal in niedriger Auflösung laufen. Es lohnt sich ein Feuerwerk in so vielen Farben anzuschauen. Versprochen....

In diesem Sinne, bis zum nächsten Mal...

SH

UVS-Software
Lernsoftware-Spezialist!
Händler-Anfragen erwünscht

Atari Special 4/88 faßt einen Artikel über Lern ST zusammen:
"Ein ideales Fremdsprachen-Lernprogramm"

Lern ST plus-Universal-Lernprogramm

Die komplett überarbeitete Version von Lern ST bietet neben zahlreichen Detailverbesserungen auch viele neue Features.

Lern ST plus ist ein Programm zum komfortablen Üben von Vokabeln und Wendungen beliebiger Sprachen. Durch Maskenkonzept läßt sich auch anderes Faktenwissen, z. B. Geschichtsdaten, lernen.
Lern ST plus enthält u. a. versch. Abfragemodi, Lexikon-, Listendruckfunktionen, Statistik und einen komfortablen Eingabeteil.

Lern ST plus kostet inkl. ausführlichem Handbuch (mit Einsteigerteil) und Updateservice auch weiterhin nur .... **DM 59,-**
Lern ST plus-Demoversion .... **DM 10,-**
Vokabeldisketten (benötig. Lern ST plus) .... je **DM 20,-**
Englisch (2200 Vokabeln)
Latein (3000 Vok. + Wendungen)
Französisch (4500 Vokabeln + Wend.)
Spanisch (4500 Vokabeln + Wend.)
Italienisch (4000 Vokabeln + Wendungen)

Deklinat/Konjugat – für Latein
Programm zum Deklinieren und Konjugieren lat. Substantive und Verben (inkl. 1680 Vokabeln), mit Anleitung .... **DM 79,-**

Neu: Quickedit ST – Editor als Accessory
Blockoperationen u.v.m., Accessory-Pipe! .... nur **DM 79,-**

Bitte fordern Sie ausführlichen Softwareprospekt an!

Ulrich Veigel Softwareservice Tel.07131/60023
Mönchseestraße 83 7100 Heilbronn
Schweiz: Data Trade, PF, CH-8021 Zürich

Professionelle Schön-Schrift
mit **Signum** und **Laser-/24-Nadeldrucker**

- Modernes, optimal lesbares Schriftbild
- in den Größen 8, 10, 12 und 15 Punkt
- einschließlich unproportionaler Ziffern
- normal und **fett**, und dazu ein schlau
- ausgetüftelter „SONDER"-Zeichensatz

mit griechischem Alphabet (Α, α, Β, β, Γ, γ, Δ, δ, ..., Ω, ω), römischen Zahlen (I, II, III, IV, ... X, ... MCMLXXXIX), fremdsprachigen (Å, å, Ç, ç, Ø, ø, Œ, œ, ñ, ¿), kaufmännischen (£, ‰, ®, ©, ...) und mathematischen Zeichen (∞, ∫, ∫, ≡, ≠, ≢, <, ≦, ≧, >, ~, ≈, ≅, ±, ×, ÷, ∢ usw.) und vielem mehr (①, ... ⑨, ⑩, ... ⑨⑨, ϟ, /, →, ←, ↑, ↓) für die unterschiedlichsten Einsatzbereiche.

Ausführliche Info mit Schriftprobe
- für 3,- DM in Briefmarken anzufordern bei:

Walter Schön, Berg-am-Laim-Str. 133a, 8000 München 80, Tel. (089) 4 36 22 31.



Für alle Calamus-DTP-Anwender!

Verschaffen Sie sich jetzt den professionellen Überblick in Ihrer DTP-Anwendung:

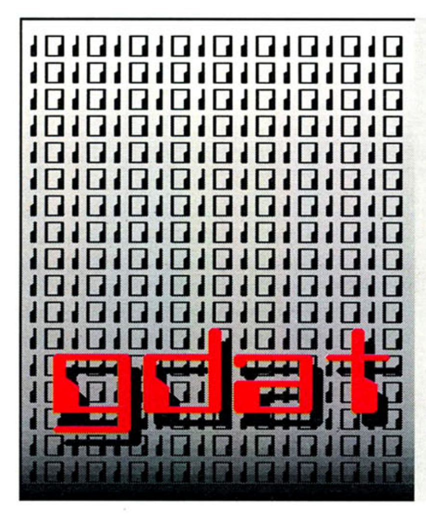
"Buchgestaltung auf Diskette für Calamus-Anwender"

Unser umfassender Einführungskurs auf 3 Disketten, randvoll mit wertvollen Informationen, vermittelt Ihnen alle nötigen Tips und Tricks aus der Welt der Typographie und der Buchgestaltung.

Hier finden Sie alles Wissenswerte aus der Welt des Buches, um Ihre Ideen und Gedanken in Ihrer alltäglichen Arbeit mit Computer und Calamus-DTP, professionell aufs Papier zu bringen.

Bestellen Sie unseren neuen DTP–Einführungskurs "Buchgestaltung auf Diskette für Calamus-Anwender", auf 3 Disketten für Atari ST-Computer, mit Begleitheft für 98.–DM bei:

Kleefeld & Partner Medienagentur, Georg-Fahrbachstr.2, 6500 Mainz 42, Tel. 06131/578489





gdat Gesellschaft für dezentrale Daten–Technik mbH
Stapelbrede 39 • 4800 Bielefeld 1 • Telefon 0521/875 888



Die Super-Grafik-Sammlung von Profis für Profis. Logos, Symbole, Pictogramme, Rahmen und Vignetten für den professionellen Einsatz mit Calamus. Hunderte von stufenlos verzerrbaren Vektorgrafiken. Vollständig überarbeitete Rastergrafiken (300 dpi), die man auch gebrauchen kann (kein Micky-Maus). Basis-Collection mit über 400 Grafiken nur DM 99.- (weitere Serien in Vorbereitung). Info anfordern oder bestellen bei:

Andreas Horn, 6000 Frankfurt 50, Hinter den Ulmen 61, Tel: 069/52 42 78, Fax: 069/53 90 96

FÜR CALAMUS!

über 600 Disketten
PD-SOFTWARE
auf TDK MF-1DD für ATARI ST & PC

für MS-DOS	ab DM 6,–
für Aladin & Sharp PC	DM 7,–
GFA-Club PD	DM 5,–
ST-PD ab Nr. 1	DM 5,–
zweiseitige DMA-PD	DM 6,–
für Signum und DTP	DM 6,–

Kopie auf Ihre Diskette gleich obige Preise minus DM 2,50
Abschlag 5,25" M2D & Aufschlag 3,5" MF-2DD je 50 Pfg.

Spiele-Pakete (farbig oder sw)	**DM 29,–**
Einsteiger-Paket	**DM 29,–**
10 TDK MF-1DD	**DM 24,–**

Porto: Vorkasse 4 DM (Inland: Nachnahme 6 DM, ab 8 Kopien frei)
Gratisinfo oder Katalogdisk mit 6 Utilities gegen 5 DM bei:

Fa. Axel Witaseck
Postfach 12 05 53
D-4000 Düsseldorf
☎ 02 11 - 23 64 99

Mengenrabatte:
ab 12 Kopien 10 %
ab 30 Kopien 15 %
ab 100 Kopien 20 %

PICTURE-DISK Grafiksammlug
1000 Grafiken DM 90.- 2400 Grafiken DM 190.-

GFA ACCESSORY-MAKER 49.–
Wandelt GFA-Basicprogramme in ACCESSORYS um !!!

ZEICHENSATZ-PROFI 2.0 119.–
Kann innerhalb von 10 Minuten einen kompletten Zeichensatz für Signum II. Dies schließt das Erzeugen von Editor- und Druckerzeichensatz ein. ZSP hat auch Funktionen wie: hohl, fett, schmal, schatten, drehen, Sinusbiegung, Freihandbiegung, Gem-Font als Sigum-Font speichern, konvertieren,. editieren, HEADLINE erzeugen, usw. INFORMATIONEN anf.

ART-PROFI FÜR SIGNUM 79.–
wie ZSP, es werden jedoch fertige Fonts benötigt.
Erzeugt aus einem Font beliebig viele neue Fonts!

CompTec Software, OT Kriwitz 12
3131 Lemgow Tel. 05883/1325 für die Schweiz:
DataTrade AG, Langstraße 94, CH-8021 Zürich





H & T

HARDWARE & TECHNIK
RAINER MERTENS HANSJOACHIM ZEGEL
1000 BERLIN 10 MIERENDORFFPLATZ 6
TEL: 030-345 30 61 FAX: 030-345 81 72

VIDEOTEXT BOX

SOFTWARE

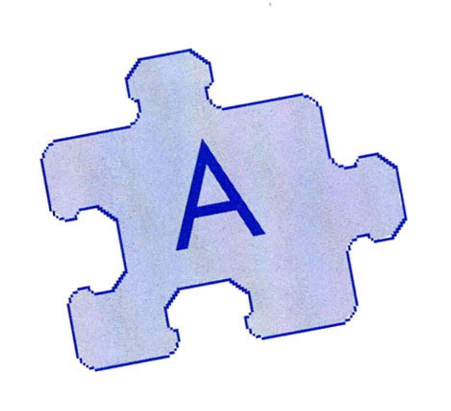
Speichern als Degas
Speichern als ASCII
Speichern Parameter
Textspeicher sichern
TOP Decoder
Seitenmemory
Mono in Graustufen
Colordarstellung

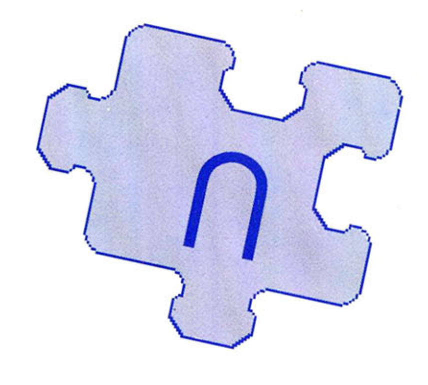
DRUCKER

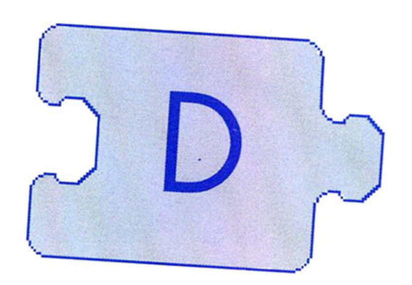
Nec P6 P7
Panasonic
Epson
HP Desk-jet
HP Laserjet II

249.-

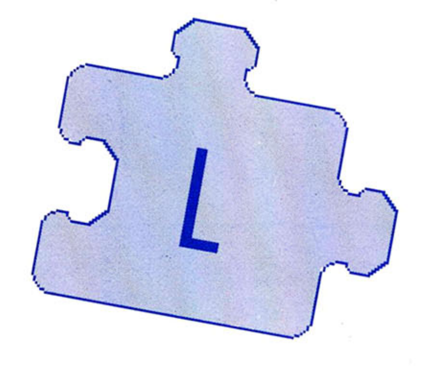


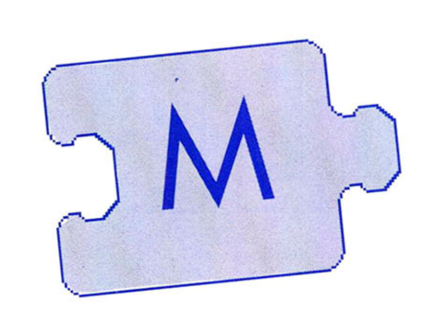






MODULA-2 KURS





Willkommen beim neunten Teil des Modula-Kurses. In dieser Folge wird das Modul-Konzept unter einem weiteren Aspekt dargestellt, dem der abstrakten Datentypen. Vorher allerdings lernen Sie einen neuen Datentyp kennen, die Zeiger.

POINTER

Wir haben inzwischen eine Reihe von Datentypen kennengelernt. Sie zeichneten sich durch zwei gemeinsame Eigenschaften aus: Mit der Variablendeklaration reserviert der Compiler automatisch einen entsprechenden Speicherbereich, und ihr Inhalt wurde direkt durch Angabe ihres Namens angesprochen. Damit zeigen sie ein statisches Verhalten: Schon zur Compile-Zeit sind die Anzahl, der Platzbedarf und der Ort (die Adresse) der Variablen bekannt.

Was wäre das Gegenteil? Welche Eigenschaften hätten dynamische Variablen? Für sie müßte während der Laufzeit des Programms Speicher eingerichtet werden. Es ließe sich eine nicht vorhersehbare Anzahl davon einrichten, und sie wären damit auch nicht mehr direkt anzusprechen, da ihre Position im Speicher erst durch die Speicheranforderung bestimmt würde.

Zunächst zur Deklaration einer dynamische Variablen. Modula-2 kennt dafür den Datentyp POINTER. Eine POINTER-Variable enthält eine Referenz auf die eigentliche Variable. Um mit dieser zu arbeiten, muß der Compiler aber noch ihren Typ kennen. Also gibt man in der Deklaration eines Zeigers an, auf welchen Typ er zeigt. Ein POINTER auf einen INTEGER wird deklariert als:

```
VAR IZeiger : POINTER TO INTEGER;
```

IZeiger ist jetzt ein Zeiger, der ein INTEGER referenziert. Es ist wichtig zu unterscheiden zwischen dem Zeiger mit Namen *IZeiger* und dem namenlosen INTEGER-Wert, auf den er zeigt.

Das Einrichten von dynamischen Variablen übernimmt das Standardmodul *Storage* mit den Routinen *ALLOCATE* und *DEALLOCATE*. Erstere reserviert einen Speicherbereich, die zweite gibt ihn wieder frei, so daß er für andere dynamische Variablen verwendet werden kann.

ALLOCATE erhält zwei Parameter: den Zeiger, der auf den Anfang des eingerichteten Speicherplatzes gerichtet wird, und die Angabe, wieviel Platz die dynamische Variable benötigt. Der Platzbedarf läßt sich einfach über die Standardfunktion *TSIZE* ermitteln. TSIZE müssen Sie eventuell aus SYSTEM importieren. Dazu ein Beispiel: Es soll eine dynamische Variable vom Typ INTEGER eingerichtet werden, die per *IZeiger* angesprochen wird (s. Listing 1).

Am Anfang steht der Import der benötigten Routinen aus Storage. Es folgt die Deklaration des Pointers *IZeiger*, der auf eine dynamische Variable vom Typ INTEGER zeigt. Im Programm kann nun die Variable eingerichtet werden. Der Aufruf von ALLOCATE richtet TSIZE(INTEGER) Bytes ein und setzt den Zeiger entsprechend.

```
MODULE ZeigerTest;
...
FROM Storage IMPORT ALLOCATE, DEALLOCATE ;
VAR IZeiger : POINTER TO INTEGER ;
BEGIN
  ...
  ALLOCATE(IZeiger, TSIZE(INTEGER));
  ...
```

Listing 1

Die zweite Routine aus Storage gibt den reservierten Speicher wieder frei. Da der tatsächliche Speicherplatz im Rechner natürlich begrenzt ist, sollte man damit nicht mehr benötigte dynamische Variablen entfernen. Und man sollte auch am Ende eines Programms entsprechend den ALLOCATE-Aufrufen DEALLOCATEs durchführen, um ein Programm zu erhalten, das sich bezüglich der Speicherverwaltung "sauber" verhält.

Im Beispiel hießen also die letzten Zeilen:

```
 ...
  DEALLOCATE(IZeiger);
END ZeigerTest.
```

Das Modul Storage kann unterschiedlich implementiert sein; Sie sollten an dieser Stelle in Ihrem Compiler-Handbuch nachschlagen, ob es weitere Routinen gibt und ob sich spezielle Hinweise zur Benutzung finden.

Das Modul "Storage" kann in einigen Systemen auch "STORAGE" heißen. Für das Public Domain-System LPR ist es nicht vorhanden. Sie finden daher am Ende des Artikels in Listing 4 eine Implementation entsprechend dem Vorschlag in Wirths Sprachdefinition.

Nun existiert die dynamische Variable; im Programm muß sie aber anders angesprochen werden als eine statische. Es geht darum, die Referenz auf die eigentliche Variable - die der Zeiger darstellt - aufzulösen; der Informatiker nennt das "Dereferenzierung". In einer Zuweisung löst das Zeichen "^" die Dereferenzierung aus. Soll dem INTEGER ein Wert zugewiesen werden, muß man schreiben:

```
IZeiger^:=42;
```

Das Objekt, auf das IZeiger zeigt, erhält den Wert 42. In Ausdrücken arbeitet die Derefenzierung analog. Mit

```
a:=IZeiger^;
```

erhielte a den Wert 42. Die Zuweisung

```
IZeiger:=42;
```

ist nicht erlaubt. Sie würde 42 nicht dem INTEGER, sondern dem POINTER zuweisen, und dabei mokiert der Compiler einen Typfehler.

Haben wir einen zweiten Zeiger *JZeiger*, der ebenfalls als POINTER TO INTEGER deklariert ist, könnte man schreiben:

```
IZeiger:=JZeiger;
```

Dabei wird allerdings nur der Zeiger zugewiesen. Beide POINTER zeigen nun auf dasselbe Objekt, nämlich den einmal im Speicher existierenden Wert 42. Die Zuweisung:

```
JZeiger^:=40;
```

ändert auch den Wert, der sich durch die Derefenzierung *IZeiger^* ergibt. *IZeiger^* liefert jetzt wie *JZeiger^* den INTEGER-Wert 40. Eine wirkliche Wertezuweisung muß lauten:

```
IZeiger^:=JZeiger^;
```

```
MODULE DynamischeVars;

FROM SYSTEM  IMPORT TSIZE;
FROM InOut   IMPORT WriteString, ReadInt, WriteInt ;
FROM Storage IMPORT ALLOCATE, DEALLOCATE ;

CONST MAXVARS = 20;
VAR n,i:INTEGER;
   Variable: ARRAY [0..MAXVARS] OF POINTER TO INTEGER;
BEGIN
  WriteString('Anzahl der Variablen ? ');
  ReadInt(n);
  FOR i:=1 TO n DO
    ALLOCATE(Variable[i],TSIZE(INTEGER));
    ReadInt(Variable[i]^);
  END;
  FOR i:=1 TO n DO
    WriteInt(Variable[i]^,5);
  END;
  FOR i:=1 TO n DO
    DEALLOCATE(Variable[i]);
  END;
END DynamischeVars.
```

Listing 2

Dabei werden die tatsächlichen INTEGER-Variablen dereferenziert, und die Zuweisung arbeitet über die tatsächlichen Werte.

Dies mag auf den ersten Blick kompliziert erscheinen, ist aber mit der oben genannten Unterscheidung zwischen dem Zeiger und dem Wert der dynamischen Variablen einfach zu verstehen. Eine Pointer-Variable ohne "^" im Programmtext bezeichnet immer nur den Zeiger und nicht den Wert. Mit "^" wird immer der Wert der dynamischen Variablen bezeichnet und nicht die Referenz darauf.

Nun ein größeres Beispiel, das demonstriert, wie die Anzahl der Variablen sich erst während der Laufzeit des Programms ergibt (s. Listing 2).

Hier kann der Benutzer bestimmen, wieviele Variablen eingerichtet werden sollen. Dazu existiert ein Feld von Zeigervariablen, das in der ersten Schleife per ALLOCATE mit gültigen Zeigern gefüllt wird. Gleichzeitig fragt das Programm in dieser Schleife nach Werten für die Variablen. Sie sehen, daß die dereferenzierte Variable als Parameter an *ReadInt* übergeben wird.

Die mittlere Schleife gibt die Eingaben auf dem Bildschirm aus. In der abschließenden Schleife sorgt das Programm für eine korrekte Speicherverwaltung, indem es per DEALLOCATE alle vorher eingerichteten Variablen wieder aufgibt.

Abstrakte Datentypen

Module, so wie sie in den letzten Folgen dargestellt wurden, stellen eine Reihe von vorprogrammierten Prozeduren samt dazugehörigen Variablendeklarationen und Typendefinitionen bereit. Aber man kann mit ihnen auch noch anders programmieren. Mit abstrakten Datentypen lassen sich Probleme noch unabhängiger von der Darstellung im Rechner modellieren, was Programme noch lesbarer und portabler macht. Doch zunächst zurück zu den Datentypen. Es gibt in Modula viele vordefinierte Typen, die Sie in Variablendeklarationen verwenden. Aber ein Datentyp ist nicht nur durch einfache Dimensionen, wie den Wertebereich eines INTEGERs, bestimmt. Die schönsten Fließkommazahlen doppelter Genauigkeit nutzen nichts ohne darauf implementierte Operationen wie z.B. der Multiplikation.

Ein Datentyp besteht also aus einer Definition der Darstellung im Rechner und Operationen auf diesem Datentypen, die irgendwo ausprogrammiert werden. Ein Modula-Compiler kennt den Datentypen INTEGER, der dem Programmierer samt einer Reihe von Operationen bereitsteht. Wie ein INTEGER dargestellt ist, weiß eigentlich nur der Compiler (obwohl man die Darstellung den Handbüchern entnehmen kann). Im Compiler sind Routinen vorprogrammiert, die immer dort eingesetzt werden, wo eine Operation auf INTEGERs im Programmtext vorkommt.

Was macht einen Datentypen nun abstrakt? Der vorherige Absatz macht eine Trennung naheliegend. Auf der einen

Seite der Typ INTEGER, der zunächst nur ein Aufzählungstyp für positive und negative ganze Zahlen ist, und die Information, daß z.B. die Operation Addition auf zwei INTEGERs möglich ist und ein Ergebnis vom Typ INTEGER hat. Auf der anderen Seite stehen konkrete Implementationskonstanten wie "ein INTEGER ist 16 Bit breit" und konkrete Code-Folgen für die Ausführung der angekündigten Operationen.

Die bekannten Informationen abstrahieren von der konkreten Implementation des Datentyps und seiner Operationen. Das ist gut so, denn wie wollten Sie ein Programm schreiben, wenn Sie bei der Benutzung einer REAL-Variablen Rücksicht darauf nehmen müßten, ob Fließkommazahlen nun im IEEE-Format oder anders dargestellt werden. Sie wollen einfach nur addieren und überlassen den Rest dem Compiler bzw. dem Laufzeitsystem.

Nehmen wir an, Sie wollen mit komplexen Zahlen umgehen. Sie kennen sie vielleicht noch aus der Schulmathematik oder aus der Elektronik: Eine komplexe Zahl beschreibt einen Punkt auf einer Zahlenfläche. Die besteht aus einem realen (entspricht der x-Achse) und einem imaginären Teil (entspricht der y-Achse). Die Mathematik definiert Rechenregeln für solche Zahlen, man kann z.B. zwei komplexe Zahlen addieren. Übrigens basieren die allseits beliebten fraktalen Darstellungen von Apfelmännchen und Seepferdchen auf komplexen Zahlen. (Falls Ihnen das Beispiel zu "komplex" oder mathematiklastig ist, dann denken Sie z.B. an die Datumsrechnung, "was ist der 3.7.89 plus 30 Tage?".)

Sie brauchen zunächst eine Definition von komplexen Zahlen, die Sie in Ihrem Programm verwenden können, und eine Reihe sinnvoller Operationen darauf. Diese müssen Sie zunächst ausprogrammieren und alles in einem Modul bereitstellen.

Und hier paßt die oben geschilderte Trennung von abstraktem und kokretem Aspekt eines Datentyps genau auf die Trennung zwischen Definitions- und Implementationsmodul. Sind die Deklarationen im Definitionsmodul so geschrieben, daß sie vollständig von der Darstellung im Implementationsmodul abstrahieren, haben Sie einen abstrakten Datentypen geschaffen.

Ein erster Versuch

Wie könnte nun unser Definitionsmodul für komplexe Zahlen aussehen? Der Anfang eines ersten Versuchs kann man im folgenden Listing sehen.

```
DEFINITION MODULE Complex;

TYPE COMPLEX = RECORD
            real,
            imag : REAL
           END;
...
```

Doch damit erhalten wir keinen abstrakten Datentypen. Warum? Schon diese Typendefinition entspricht nicht der geforderten Abstraktion von der Darstellung eines Datentyps. Im Gegenteil: Die Definition beschreibt nichts anderes als die konkrete Implementierung des Typs. Daß es sich um einen RECORD handelt und daß dessen Felder *real* und *imag* heißen, soll eben nicht bekannt werden. Ebenso soll niemand wissen, daß beide als REAL dargestellt werden. Was wir brauchen ist:

```
DEFINITION MODULE Complex;

TYPE COMPLEX;
...
```

"Was ist das?", wird der aufmerksame Leser fragen. Wir setzen zu einer Typdefinition an, brechen dann aber ab, ohne zu schreiben, wie die Definition denn eigentlich aussieht. In Modula ist obiges Fragment völlig legal, denn es handelt sich um einen speziellen Export, den "opaquen" Export.

Bei einem solchen Export, der nur für Typen möglich ist, wird im Definitionsmodul die eigentliche Definition weggelassen, lediglich der Bezeichner ist als Typ nach außen bekannt. Seine tatsächliche Definition steht im Implementationsmodul.

Daraus ergibt sich für den Compiler aber ein Problem. Das Definitionsmodul wird bekanntlich separat vom Implementationsmodul compiliert. Bei einem IMPORT liest der Compiler nur das übersetzte Definitionsmodul ein.

Findet ein opaquer Export statt, gibt es an diesem Punkt keinerlei Informationen über den Platzbedarf des neuen Typs, wodurch eine korrekte Übersetzung mit Platzreservierung für die statischen Variablen unmöglich wäre.

Aus diesem Dilemma hilft eine Einschränkung, die allerdings nur auf den ersten Blick als sehr streng erscheint. Opaque Typen dürfen nur als POINTER implementiert werden. Damit ist ihr Platzbedarf stets gleich. Da allerdings mit einem Zeiger alle dynamischen Daten beliebigen Typs angesprochen werden können, ist die Einschränkung nicht mehr so hart.

Bei einem opaquen Typen steht im Definitionsmodul also immer "TYPE <Bezeichner>" und im Implementationsmodul "TYPE <Bezeichner> = POINTER TO ...". Bei der anstehenden Revision von Modula-2 könnte diese Beschränkung übrigens fallen.

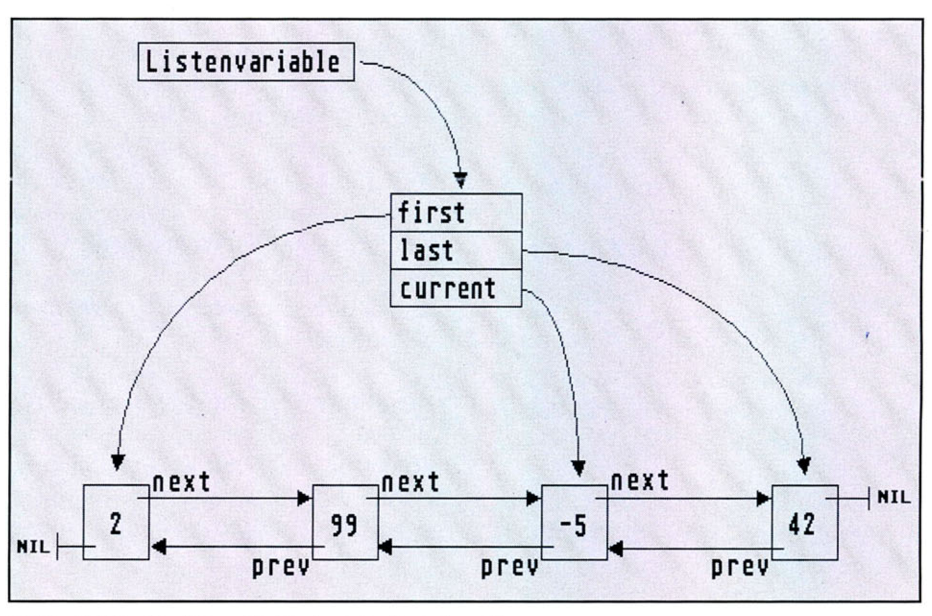


Bild 1: Aufbau der Listen, die implementiert werden sollen.

Zurück zum Beispiel der komplexen Zahlen. Listing 5 zeigt das komplette Definitionsmodul. Es wird der Typ COMPLEX samt den dazugehörigen Operationen exportiert.

Im Implementationsmodul findet sich die tatsächliche Definition von COMPLEX: Es ist ein Zeiger auf einen Record, der die beiden einer Bestandteile der komplexen Zahl aufnehmen kann.

Von den Prozeduren sind fünf Standardoperationen für opaque Typen. *MakeComplex* richtet den Record per ALLOCATE ein und setzt damit den Zeiger. *KillComplex* entfernt den Record aus dem Speicher und setzt sicherheitshalber den Zeiger auf NIL.

Was ist NIL? NIL ist ein vordefinierter Wert für nicht gültige Zeiger. Immer, wenn eine Zeigervariable den Wert NIL hat, darf sie nicht dereferenziert werden, das Laufzeitsystem wird sogar einen Fehler melden.

Real, *Imag* und *SetComplex* sind die direkten Zugriffsoperationen für COMPLEX. Sie liefern den realen und imaginären Teil zurück bzw. setzen die Variable. Direkte Zugriffe sind ja nicht möglich, da die Implementierung des Typs nicht nach außen bekannt ist. Daher müssen diese Zugriffsoperationen bei jedem abstrakten Datentypen vorhanden sein.

Die restlichen Prozeduren schließlich implementieren die mathematischen Operationen für komplexe Zahlen. Die Formeln stammen aus der Mathematik und sollen hier nicht näher erläutert werden.

Die Implementierung ist übrigens mit LPR-Modula entwickelt worden. Die Konstante *pi*, die hier aus MathCom importiert wird, kann in anderen Systemen auch in MathLib0 stehen. Eine Anpassung sollte Ihnen nach Handbuchstudium keine Probleme machen.

Listen

Das dritte Modul in dieser Folge soll eine wichtige Datenstruktur implementieren, die Listen. Sie zeichnen sich dadurch aus, daß in ihnen dynamisch Werte eingetragen werden können und keine Begrenzung wie bei den statischen Feldern besteht. Das Modul kombiniert die zwei Hauptthemen dieser Folge, die Zeiger und abstrakte Datentypen.

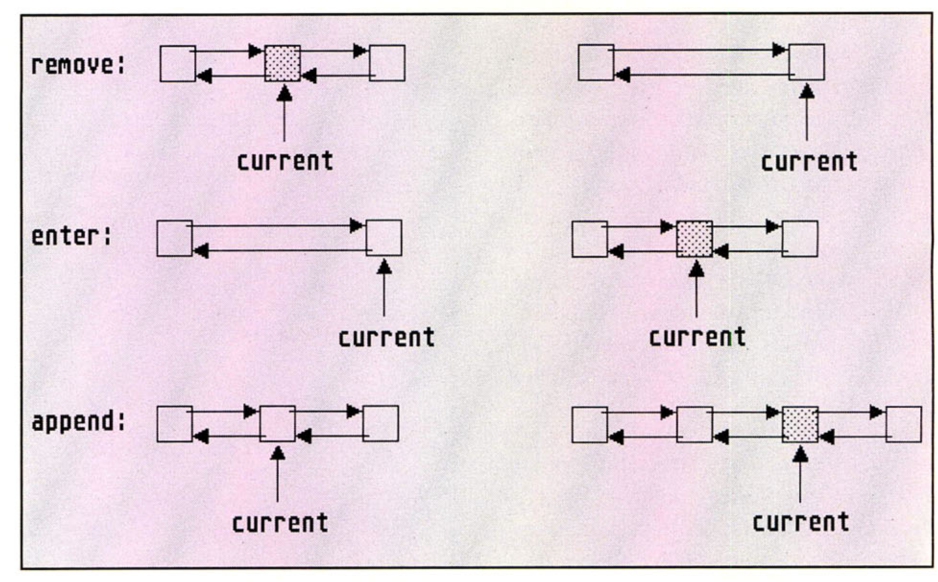


Bild 2: Grafische Darstellung der Zeigeroperationen

Eine Liste ist eine Struktur im Speicher, deren Elemente untereinander mit Zeigern verbunden sind. Die Verzeigerung hält das Ganze zusammen, über sie sind alle Elemente der Liste zu erreichen. Bild 1 zeigt den Aufbau der Listen, die hier implementiert werden sollen.

Im unteren Teil sehen Sie die Listenelemente, die aus drei Teilen bestehen - sie werden als RECORDs implementiert und dynamisch eingerichtet. Ein Feld dient der Aufnahme eines Wertes, also den eigentlichen Daten. In unserem Fall sind dies INTEGERs. Weiterhin hat jedes Listenelement zwei Zeiger, *next* und *prev*.

Sie zeigen in der Listenstruktur auf das Nachfolger- und Vorgängerelement. Sind die Zeiger korrekt gesetzt, kann man sich mit ihrer Hilfe durch die Liste bewegen. Im Beispiel kann man durch zweifache Dereferenzierung der *prev*-Zeiger vom Element mit dem Wert 5 auf das mit dem Wert -99 kommen. Eine solche Liste mit Zeigern in beide Richtungen nennt man "zweifach-verzeigert".

Wie sieht es am Anfang und Ende der Liste aus? Die dortigen Listenelemente haben keinen Vorgänger bzw. Nachfolger. Dementsprechend sind *prev* bzw. *next* undefiniert und sollen daher den Wert NIL haben. Besteht die Liste nur aus einem Element, dann sind bei diesem beide Zeiger NIL.

Soweit die Struktur innerhalb der Liste. Die Elemente liegen dynamisch im Speicher und sind untereinander vollständig verzeigert. Es gibt aber noch keine Möglichkeit, die Elemente von außen anzusprechen. Dazu dient ein RECORD *ListHeader*, der die Liste zusammenhält und erreichbar macht.

In dieser Implementierung besteht er aus drei Zeigern. Der erste, *first*, eröffnet den Zugriff auf die Liste. Er zeigt auf das erste Listenelement, und nachdem man von einem Element zu allen anderen gelangen kann, bekommt man durch Dereferenzierungen beginnend bei *first* Zugriff auf alle Werte. In dem Beispiel im Bild liefert der Ausdruck "first^.next^.next^" den Wert -5.

Der Zeiger last ist aus Effizienzgründen vorhanden; er zeigt auf das letzte Element der Liste. Natürlich wäre es auch durch die Abfrage "next=NIL" zu erkennen, dazu müßte man sich aber durch die gesamte Liste bei *first*^ beginnend durchhangeln. An einigen Stellen der Implementierung macht sich diese Effizienz bemerkbar.

Schließlich gibt es noch einen dritten Zeiger, *current*. Er stammt aus Überlegungen zu den Operationen, die für die Listen implementiert werden sollen. Sie wirken auf ein Element der Liste, nur kann dieses ja nicht direkt angesprochen werden, da es sich um einen abstrakten Datentypen handelt. Daher wird current eingerichtet, ein Zeiger, der mit Operationen vor- und rückwärts in der Liste bewegt wird oder direkt an den Anfang und das Ende der Liste gesetzt werden kann. Operationen, wie z.B. das Auslesen des Wertes, beziehen sich immer auf das Element, auf das *current* zeigt.

Schließlich soll das Listenmodul einen abstrakten Datentypen implementieren.

Daher soll ein direkter Zugriff auf *ListHeader* nicht erlaubt sein. Das Benutzerprogramm kennt nur eine Listenvariable, die ein Zeiger auf eine *ListHeader*-Struktur ist. Diese wiederum wird für jede angeforderte Liste dynamisch eingerichtet.

Damit ist die Datenstruktur beschrieben, auf der das Modul arbeiten soll. In Listing 6 finden Sie das Definitionsmodul *INTLists*, das die vorhandenen Operationen auflistet.

Aus Platzgründen möchte ich hier nur auf die zwei Operationen eingehen, die viel an der Zeigerstruktur der Liste manipulieren, nämlich *EnterElement* und *RemoveElement*.

EnterElement soll ein weiteres Listenelement vor dem aktuellen (durch *current* bezeichnet) in die Liste einfügen und den aktuellen Zeiger auf das neue Element setzen. Dazu müssen insgesamt fünf Zeiger gesetzt werden.

Das aktuelle Element soll zum Nachfolger des neuen werden. Der Vorgänger des aktuellen Elements soll als Nachfolger das neue haben. Zunächst muß dazu das neue Element per ALLOCATE eingerichtet werden. Im Listing wird dessen Speicherlage in p festgehalten.

Um dem Vorgänger des alten aktuellen Elements das neue als Nachfolger bekanntzumachen, muß dessen *next*-Zeiger auf p gesetzt werden. Damit die Verzeigerung stimmt, erhält der *prev*-Zeiger von *p* den Wert des *prev*-Zeigers des aktuellen Elements. Somit stimmt die Verzeigerung vom neuen Element zum ehemaligen Vorgänger des aktuellen Listeneintrags.

Der Nachfolger des neuen Elements soll das momentan noch aktuelle Element sein. Dazu müssen zwei Zeiger verändert werden. *current^.prev* zeigt auf *p*, das ja nun der neue Vorgänger von *current* sein soll. Die doppelte Verzeigerung stimmt wieder, wenn *p*^.next auf *current* zeigt.

In diesem Moment ist die Verzeigerung wieder komplett und das neue Element in die Liste eingefügt. Entsprechend der Definition der *EnterElement*-Operation wird schließlich noch *current* auf das neue Element *p* korrigiert. Die Zuweisungen im Implementations-Modul entsprechen diesen Zeigeroperationen. Sie finden sie in Bild 2 nochmals grafisch dargestellt.

```
DEFINITION MODULE Storage;
(* 3.7.89 Robert Tolksdorf, initialimplementation *)
FROM SYSTEM IMPORT ADDRESS ;

PROCEDURE ALLOCATE(VAR a:ADDRESS; size:LONGCARD);
PROCEDURE DEALLOCATE(VAR a:ADDRESS);
PROCEDURE Available(size:LONGCARD): BOOLEAN;

END Storage.


IMPLEMENTATION MODULE Storage;
(* 3.7.89 Robert Tolksdorf, initialimplementation *)
FROM SYSTEM IMPORT ADDRESS, VAL ;
FROM GEMDOS IMPORT Alloc, Free;

PROCEDURE ALLOCATE(VAR a:ADDRESS; size:LONGCARD);
BEGIN
  IF Available(size) THEN
    Alloc(size, a);
  ELSE
    HALT
  END;
END ALLOCATE;

PROCEDURE DEALLOCATE(VAR a:ADDRESS);
VAR done:BOOLEAN;
BEGIN
  done:=Free(a);
END DEALLOCATE;

PROCEDURE Available(size:LONGCARD): BOOLEAN;
VAR a:ADDRESS;
BEGIN
  Alloc(-1,a);
  RETURN(VAL(LONGCARD,a)>=size);
END Available;
END Storage.
```

```
DEFINITION MODULE Complex;

TYPE COMPLEX;

PROCEDURE MakeComplex(VAR a:COMPLEX);
PROCEDURE KillComplex(VAR a:COMPLEX);
PROCEDURE Real(a:COMPLEX):REAL;
PROCEDURE Imag(a:COMPLEX):REAL;
PROCEDURE SetComplex(a:COMPLEX; r,i:REAL);
PROCEDURE AddComplex(a,b:COMPLEX; VARresult:COMPLEX);
PROCEDURE SubComplex(a,b:COMPLEX; VARresult:COMPLEX);
PROCEDURE MulComplex(a,b:COMPLEX; VARresult:COMPLEX);
PROCEDURE DivComplex(a,b:COMPLEX; VARresult:COMPLEX);
PROCEDURE AbsComplex(a:COMPLEX):REAL;
PROCEDURE ArgComplex(a:COMPLEX):REAL;

END Complex.


IMPLEMENTATION MODULE Complex;

FROM MathLib0   IMPORT sqrt, arctan ;
FROM MathCom    IMPORT pi ;
FROM Storage    IMPORT ALLOCATE, DEALLOCATE ;
FROM SYSTEM     IMPORT TSIZE;

TYPE COMPLEX = POINTER TO complex;
     complex = RECORD
                 real,imag : REAL
               END;

PROCEDURE MakeComplex(VAR a:COMPLEX);
BEGIN
  ALLOCATE(a,TSIZE(Complex));
END MakeComplex;

PROCEDURE KillComplex(VAR a:COMPLEX);
BEGIN
  DEALLOCATE(a);
  a:=NIL;
END KillComplex;

```

→

Bei der Operation *RemoveElement* soll das aktuelle Element aus der Liste entfernt werden. Dazu muß der Vorgänger von *current* zum direkten Vorgänger von *current^.next* werden und der Nachfolger von *current* zum direkten Nachfolger von *current^.prev*. Die Zeiger "übergehen" praktisch das aktuelle Element.

Die Implementierung ist recht einfach. Man braucht lediglich den *next*-Zeiger des Vorgängers und den *prev*-Zeiger des Nachfolgers von *current* durch Zuweisung zu setzen und kann somit das momentan noch aktuelle Element per DEALLOCATE aus dem Speicher entfernen. Übrig bleibt noch das Aktualisieren von *current* auf dessen Nachfolger.

Alle Routinen benötigen eine besondere Behandlung der Zeigerstruktur, falls die Liste leer oder das aktuelle Element das erste oder letzte der Liste ist. Dabei müssen die *first*- und *last*-Zeiger aktualisiert und der Zugriff auf nicht vorhandene Nachfolger und Vorgänger verhindert werden.

Im Listing werden Sie einige auskommentierte Zeilen finden, die eigentlich korrektes Modula-2 sind. Es hat sich jedoch herausgestellt, daß die ETH-Compiler (z.B. LPR oder SPC) Probleme haben, wenn in einem Ausdruck eine dreifache Dereferenzierung vorkommt. Durch eine Zwischenvariable (*p* oder *q*) umgeht das Modul diese unschöne Implementationsbeschränkung.

Die heutige Beschreibung komplexer Datenstrukturen kann das Thema aus Platzgründen nur ansatzweise anreißen, auch ist die Beschreibung der Routinen sicherlich nicht sehr detailreich. Ich möchte Ihnen daher zwei Empfehlungen zum Weiterlesen geben. Falls Sie ältere Exemplare von ST Computer besitzen, lohnt sich ein Blick in die Serie "Algorithmen und Datenstrukturen in Pascal" von Dirk Brockhaus in ST Computer 10/87 bis 10/88.

Die ultimative Quelle für den Modula-2-Programmierer ist jedoch das Buch "Algorithmen und Datenstrukturen in Modula-2" vom Modula-Schöpfer Niklaus Wirth. Erschienen ist das Buch beim B.G. Teubner Verlag, Stuttgart, 1986, in der vierten Auflage. Im Buchhandel können Sie es unter der ISBN 3-519-02260-5 für DM 38,- erwerben. Wollen Sie ernsthaft mit Modula-2 arbeiten, ist das Geld wirklich gut angelegt.

```
PROCEDURE Real(a:COMPLEX):REAL;
BEGIN
  RETURN a^.real;
END Real;

PROCEDURE Imag(a:COMPLEX):REAL;
BEGIN
  RETURN a^.imag;
END Imag;

PROCEDURE SetComplex(a:COMPLEX; r,i:REAL);
BEGIN
  a^.real:=r;
  a^.imag:=i;
END SetComplex;

PROCEDURE AddComplex(a,b:COMPLEX; VARresult:COMPLEX);
BEGIN
  result^.real:=a^.real+b^.real;
  result^.imag:=a^.imag+b^.imag;
END AddComplex;

PROCEDURE SubComplex(a,b:COMPLEX; VARresult:COMPLEX);
BEGIN
  result^.real:=a^.real-b^.real;
  result^.imag:=a^.imag-b^.imag;
END SubComplex;

PROCEDURE MulComplex(a,b:COMPLEX; VARresult:COMPLEX);
BEGIN
  result^.real:=a^.real*b^.real-a^.imag*b^.imag;
  result^.imag:=a^.real*b^.imag+a^.imag*b^.real;
END MulComplex;

PROCEDURE DivComplex(a,b:COMPLEX; VARresult:COMPLEX);
BEGIN
  result^.real:=(a^.real*b^.real+a^.imag*b^.imag)/
                (b^.real*b^.real+b^.imag*b^.imag);
  result^.imag:=(a^.imag*b^.imag-a^.real*b^.imag)/
                (b^.imag*b^.real-b^.real*b^.imag);
END DivComplex;

PROCEDURE AbsComplex(a:COMPLEX):REAL;
BEGIN
  RETURN(sqrt(a^.real*a^.real+a^.imag*a^.imag));
END AbsComplex;

PROCEDURE ArgComplex(a:COMPLEX):REAL;
BEGIN
  IF (a^.real=0.0) AND (a^.imag=0.0) THEN
    RETURN 0.0
  ELSE
    IF ABS(a^.real)>=ABS(a^.imag) THEN
      IF (a^.real>0.0) THEN
        RETURN arctan(a^.imag/a^.real)
      ELSE
        IF (a^.imag>=0.0) THEN
          RETURN arctan(a^.imag/a^.real)+pi
        ELSE
          RETURN arctan(a^.imag/a^.real)-pi
        END
      END
    ELSE
      IF (a^.imag>0.0) THEN
        RETURN pi/2.0-arctan(a^.real/a^.imag)
      ELSE
        RETURN -pi/2.0-arctan(a^.real/a^.imag)
      END
    END
  END
END ArgComplex;

END Complex.
```

STORAGE für LPR-Modula

An dieser Stelle das versprochene Modul Storage für LPR-Modula, das dem PD-System bisher fehlte. Die Implementierung des Moduls greift auf die GEMDOS-Routinen *Alloc* und *Free* zurück. Erstere reserviert einen Speicherbereich, falls die Anzahl der zu reservierenden Bytes positiv ist. Sie liefert die Anzahl der freien Bytes zurück, falls eine -1 übergeben wird. *Free* gibt den vorher an der übergebenen Adresse reservierten Speicher dem

Betriebssystem zurück. ALLOCATE und DEALLOCATE werden direkt auf diese Aufrufe abgebildet. ALLOCATE löst einen HALT aus, falls die Anforderung zu groß war.

Das kleine Modul ist ein gutes Beispiel dafür, wie von einem konkreten Betriebssystem - hier GEMDOS - abstrahiert wird, indem man ein von der Sprachbeschreibung vorgeschlagenes Standardmodul implementiert. Wirth schlägt übrigens einen CARDINAL-Typen für size vor; dies dürfte aber ein Relikt aus Zeiten sein, in denen 16 Bit die Breite des Adreßbusses bei Mikro-Computern war (Beim Public Domain-System von Fitted Software-Tools für MS-DOS und PC-DITTO bzw. pc-speed wird ebenfalls ein CARDINAL verwendet, was sich aus der möglichen Segmentierung des Speichers in 64kByte-Seiten bei Intel-Prozessoren ergibt).

Sie erkennen an der Implementierung übrigens eine Eigenschaft des Typs ADDRESS aus SYSTEM: Er ist zu allen POINTER-Typen kompatibel.
Es gibt Implementierungen, die zusätzlich eine Variable exportieren, die das Verhalten von ALLOCATE bestimmt, falls nicht genug Speicher verfügbar ist. Man könnte in diesem Fall NIL zurückliefern oder eine Variable done exportieren, die den Erfolg der Speicheranforderung und damit die Gültigkeit des Zeigers signalisiert.

Abschließend darf ich Sie auf die erweiterten und zusätzlichen Module zum LPR-System hinweisen, die mir von Georg Galster von der dänischen ST-User-Group ST/OP - The Danish Society of Atari ST/OPerators - zugeschickt wurden. Nachdem er uns freie Verwendbarkeit zugesichert hat, sind sie in die Public Domain-Sammlung von ST Computer aufgenommen worden.

An dieser Stelle, da sich der Kurs dem Ende nähert, darf ich die geneigten Leser nochmals auffordern, offen gebliebene Fragen zu Modula-2 zu stellen. Die letzte Folge soll in einer (in Amerika recht beliebten) Art Frage-Antwort-System die restlichen Aspekte von Modula-2 klären. Zuschriften bitte an die ST-Computer-Redaktion zu Händen R. Tolksdorf! Fragen zum LPR-System werde ich ebenfalls beantworten; falls der Platz nicht reicht, in der Schwesterzeitschrift PD-NEWS.

In der nächsten Folge kommen wir zu den Möglichkeiten, mit Modula-2 nebenläufige und parallele Programme zu schreiben.

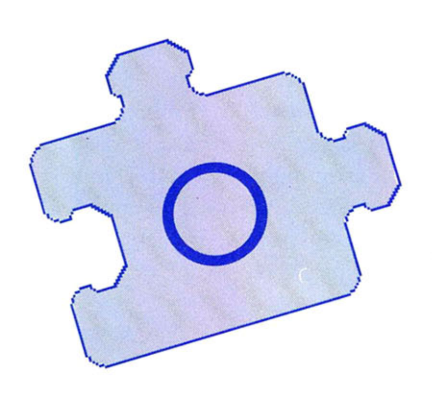
RT

Hausaufgaben

1. Schreiben Sie eine Prozedur *NewInt*, die eine dynamische INTEGER-Variable erzeugt und sie auf einen übergebenen Wert setzt. Sie hat zwei Parameter, nämlich einen INTEGER-Wert und den zu setzenden Zeiger.

2. Schreiben Sie ein kleines Modul, das die aus den früheren Aufgaben bekannte Ampel als abstrakten Datentypen bereitstellt. Es sollen fünf Prozeduren implementiert werden, die eine Ampel-Variable einrichten und löschen, sie auf einen Wert setzen, den Wert auslesen und die Ampel "weiterschalten". Den Typ *AmpelFarben* sollten Sie als Aufzählungstyp exportieren.

3. Schreiben Sie unter Verwendung des INTLists-Moduls ein kleines Programm, das eine beliebige Anzahl von Werten als Eingabe erhält und daraus die Summe sowie den Mittelwert bildet.



```
DEFINITION MODULE INTLists;
  (* Der abstrakte Datentyp List *)
TYPE List;
  (* Eine Liste einrichten *)
PROCEDURE MakeList(VAR L:List);
  (* Eine Liste entfernen *)
PROCEDURE KillList(VAR L:List);
  (* Erstes Listenelement wird aktuelles *)
PROCEDURE First(L:List);
  (* Letztes Listenelement wird aktuelles *)
PROCEDURE Last(L:List);
  (* Nächstes Listenelement wird aktuelles *)
PROCEDURE Next(L:List);
  (* Vorheriges Listenelement wird aktuelles *)
PROCEDURE Prev(L:List);
  (* TRUE, wenn die Liste leer *)
PROCEDURE Empty(L:List):BOOLEAN;
  (* TRUE, wenn aktuelles gleich erstes
     Listenelement *)
PROCEDURE AtFirst(L:List):BOOLEAN;
  (* TRUE, wenn aktuelles gleich letztes
     Listenelement *)
PROCEDURE AtLast(L:List):BOOLEAN;
  (* Liefert den Inhalt des aktuellen
     Listenelements *)
PROCEDURE GetValue(L:List):INTEGER;
  (* Setzt den Inhalt des aktuellen Listenelements
     *)
PROCEDURE SetValue(L:List; value:INTEGER);
  (* Fügt ein neues vor das aktuelle Listenelement
     ein.
     Neues Element wird zum aktuellen *)
PROCEDURE EnterElement(L:List);
  (* Fügt ein neues nach dem aktuellen
     Listenelement ein.
     Neues Element wird zum aktuellen *)
PROCEDURE AppendElement(L:List);
  (* Entfernt das aktuelle Listenelement *)
PROCEDURE RemoveElement(L:List);

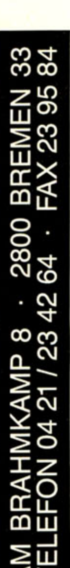
END INTLists.


IMPLEMENTATION MODULE INTLists;

(* Version*: 08.07.89   17:15         *)
FROM SYSTEM  IMPORT TSIZE ;
FROM Storage IMPORT ALLOCATE, DEALLOCATE;

```

→



FOTOSATZ

QUALITÄT mit *Calamus* in 1270 dpi

Wir belichten Ihre Calamus-Dokumente mit Linotype-Laserbelichter 300.

Diese Anzeige wurde mit Calamus/Atari erstellt und in 1270 dpi auf Fotosatz-Film belichtet.

Informieren Sie sich bei uns!



ATARI ST SOFTWARE

AS-HAUSHALT (Haushaltsbuchführung) **DM 99.-**
Buchführung für alle Privathaushalte. Buchen über Konto/Gegenkonto. Suchen, Löschen und Ändern von Buchungen. Auto-Save. Monats- und Jahresabschluß. Universelle Druckeranpassung. Journalausdruck. Statistik als Balkengrafik. Deutsches Handbuch.

ST-FIRMENBUCH (Buchführung für Gewerbetreibende) **DM 139.-**
Doppelte Buchführung für Gewerbetreibende. Automatisches Mitführen der MwSt.-Konten. Fünf versch. MwSt-Sätze frei einstellbar. Universelle Druckeranpassung. Abschlußzeitraum Monat, Quartal oder Jahr. Einnahmen/Überschußrechnung. Ausgabe eines Journals. Umsatzsteuervoranmeldung. Ausführliches deutsches Handbuch.

ST-ÜBERWEISUNGSDRUCK **DM 45.-**
Bedruckt alle Arten von Überweisungsträgern, Schecks, Zahlkarten usw. Universelle Druckeranpassung, eigene Formulare können mit einem Texteditor selbst erstellt bzw. angepaßt werden. Umwandlung des Betrages in ein Zahlwort. Die Eingaben sind speicherbar.

DATENBANKANWENDUNGEN **DM 69.-**
6 Anwendungen für Adimens ST, z.B. Termine/Adressen, Lager, Videocassetten, Schülernoten, Zeitschriftenartikel, Bibliothek.

ST-SCHREIBMASCHINE **DM 59.-**
Mit diesem Programm arbeitet Ihr Atari ST + Matrixdrucker wie eine Schreibmaschine. Sehr gut geeignet zum Ausfüllen von amtlichen Formularen. Ausdruck entweder direkt oder über editierbares Display. 10 Floskeltasten definierbar. Druckeranpassung. Deutsche Anleitung.

ST-ETIKETT Bedruckt alle Arten von Etiketten. Ausgabebild selbst anpaßbar. Mit Seriennummerngenerierung. **DM 59.-**

***NEU:* ST-TRAINER MATHEMATIK** **DM 79.-**
Lern- und Trainingsprogramm für Schüler (1. bis 5. Schuljahr). Abfrage der vier Grundrechenarten, Bruchrechnen, Kürzen, Erweitern, Längen-, Flächen-, Raum- und Gewichtsmaße (Umrechnungen). Die Abfrage- und Benotungsbereiche sind frei einstellbar. Protokolldruck optional. GEM. Mit deutschem Handbuch.

INFO 1/89 kostenlos!
Lieferung per Vorauskasse (V-Scheck) oder per Nachnahme (plus DM 5.- Versandkosten)!

AS DATENTECHNIK
AS-DATENTECHNIK, MAINZER STR.69, D-6096 RAUNHEIM. Tel.: 06142 / 2 26 77

ATARI ST SOFTWARE

Sie suchen ein leistungsfähiges Programm, mit dem Sie Ihre gesamte Auftragsabwicklung durchführen können???
Dann sollten Sie

ST-AUFTRAG

unbedingt kennenlernen!!!

ST-AUFTRAG ist ein Programm zur Auftragsbearbeitung (Angebote, Auftragsbestätigungen, Lieferscheine, Rechnungen, Bestellungen, Gutschriften) mit integrierter Artikel- und Adressenverwaltung. Der Dialog zwischen dem Anwender und ST-Auftrag wird über die komfortable Benutzeroberfläche GEM geführt. Zusätzlich sind alle wichtigen Funktionen über Tastaturkommandos erreichbar. ST-Auftrag enthält weder Kopierschutz, Hardwareschlüssel noch sonstige Anwenderhindernisse. Die Installation des Systems auf Festplatte in jedem beliebigen Ordner ist problemlos möglich.

Leistungsbeschreibung in Stichworten:

Max. Anzahl von Artikeln und Adressen: 65535; Artikellisten und -etiketten; Adressenlisten, -etiketten und Serienbrieferstellung; Offene-Posten-Liste; Bestellvorschlagsliste; autom. Mahnwesen; Lagerüberwachung; Rabattliste für jeden Artikel separat; alle Ausgaben auf Drucker, Bildschirm oder Datei; Ausgaben mit 1st Word Plus Formularen (d.h. alle Schriftattribute, Schriftbreiten, Kopf- und Fußzeilen werden mit ausgegeben); Originalgetreue Ausgabe auf Bildschirm; Artikel mit Langtext (ab Version 1.10); Barverkauf; 10 frei eingebbare Liefer/Versandarten; Ausfüllen von Nachnahmezahlkarten...

Hardwarevoraussetzungen:

Monochrommonitor, doppelseitiges Diskettenlaufwerk, Festplatte empfehlenswert, ATARI ST oder MEGA ST mit mindestens 1MB RAM.

Der Preis:

nur DM 398.-

Und damit Sie ST-AUFTRAG ausgiebig testen können, gibt es eine Demoversion, die keinerlei Funktionsbeschränkungen hat (max. Anzahl Adressen 2, Artikel 5). Die Demoversion kostet DM 20.- (mit Handbuch DM 50.-). Die Preise für Demoversion und Handbuch werden beim Kauf voll angerechnet!

Versand per Vorauskasse oder per Nachnahme (plus DM 5.-).

AS DATENTECHNIK
AS-DATENTECHNIK, MAINZER STR.69, D-6096 RAUNHEIM. Tel.: 06142 / 2 26 77

SCSI - Festplatten für ATARI ST Computer

OMEGA H85:
85 MB Seagate SCSI-Platte 28 ms

- Bootfähig
- voll kompatibel (Aladin,R-TOS,PC-Ditto fähig)
- Schnell, leise und preiswert
- DMA-Bus durchgeschliffen
- SCSI-Bus herausgeführt
- Zweite Harddisk im Gehäuse möglich
- Autopark Hardwaremäßig

nur 1769.- DM

Händleranfragen erwünscht



Computer Systeme GmbH

Oeltzenstr. 14
3000 Hannover 1
Tel. 0511-15768/15725
Fax 0511-17757

```
TYPE List               = POINTER TO ListHeader;
     ListElementPointer = POINTER TO ListElement;
     ListHeader         = RECORD
                            current,
                            first,last:
                            ListElementPointer;
                          END;
     ListElement        = RECORD
                            next,prev:
                            ListElementPointer;
                            value    : INTEGER;
                          END;

PROCEDURE MakeList(VAR L:List);
BEGIN
  ALLOCATE(L, TSIZE(ListHeader)); (* Header
                                   einrichten  *)
  L^.first:=NIL;              (* Alle Zeiger
                                 undefiniert *)
  L^.last :=NIL;
  L^.current:=NIL;
END MakeList;

PROCEDURE KillList(VAR L:List);
VAR p,q:ListElementPointer;
BEGIN
  p:=L^.first;      (* Alle Listenelemente
                       entfernen *)
  WHILE (p#NIL) DO
    q:=p;           (* Zeiger merken         *)
    p:=p^.next;     (* eins weiter           *)
    DEALLOCATE(q);  (* altes p entfernen     *)
  END;
  DEALLOCATE(L);    (* Header entfernen      *)
  L:=NIL;           (* Liste ist undefiniert *)
END KillList;

PROCEDURE First(L:List);
BEGIN
  L^.current:=L^.first;
END First;

PROCEDURE Last(L:List);
BEGIN
  L^.current:=L^.last;
END Last;

PROCEDURE Next(L:List);
BEGIN
  (* current und current^.next gültig ? *)
  IF (~Empty(L) AND (L^.current^.next#NIL)) THEN
    L^.current:=L^.current^.next;
  END;
END Next;

PROCEDURE Prev(L:List);
BEGIN
  (* current und current^.prev gültig *)
  IF (~Empty(L) AND (L^.current^.prev#NIL)) THEN
    L^.current:=L^.current^.prev;
  END;
END Prev;

PROCEDURE Empty(L:List):BOOLEAN;
BEGIN
  RETURN L^.first=NIL;
END Empty;

PROCEDURE AtFirst(L:List):BOOLEAN;
BEGIN
  RETURN L^.current=L^.first;
END AtFirst;

PROCEDURE AtLast(L:List):BOOLEAN;
BEGIN
  RETURN L^.current=L^.last;
END AtLast;

PROCEDURE GetValue(L:List):INTEGER;
BEGIN
  IF ~Empty(L) THEN (* current gültig ? *)
    RETURN L^.current^.value
  ELSE
    RETURN -1;     (* Fehlerwert *)
  END;
```

```
END GetValue;

PROCEDURE SetValue(L:List; value:INTEGER);
BEGIN
  IF ~Empty(L) THEN (* ist current gültig ? *)
    L^.current^.value:=value;
  END;
END SetValue;

PROCEDURE EnterElement(L:List);
VAR p,q:ListElementPointer;
BEGIN
  ALLOCATE(p,TSIZE(ListElement));
  IF Empty(L) THEN      (* leere Liste, alle
                           Zeiger setzen *)
    L^.first:=p;
    L^.last:=p;
    p^.next:=NIL;
    p^.prev:=NIL;
  ELSIF AtFirst(L) THEN     (* an den Anfang,
                               first setzen *)
    p^.next:=L^.first;
    L^.first:=p;
    p^.prev:=NIL;
    L^.current^.prev:=p;
  ELSE
    p^.next:=L^.current;                (* p
                                 zwischenhängen *)
    p^.prev:=L^.current^.prev;
(*    L^.current^.prev^.next:=p;*)
    q:=L^.current^.prev; (* p als neuen
                          Nachfolger setzen *)
    q^.next:=p;
    L^.current^.prev:=p;  (* p als neuen
                          Vorgänger setzen *)
  END;
  L^.current:=p;                          (*
                            current setzen *)
END EnterElement;

PROCEDURE AppendElement(L:List);
VAR p,q:ListElementPointer;
BEGIN
  ALLOCATE(p,TSIZE(ListElement));
  IF Empty(L) THEN      (* Leere Liste, alle
                           Zeiger setzen *)
    L^.first:=p;
    L^.last:=p;
    p^.next:=NIL;
    p^.prev:=NIL;
  ELSIF AtLast(L) THEN             (* Am Ende,
                                last setzen *)
    p^.prev:=L^.last;
    L^.last:=p;
    p^.next:=NIL;
    L^.current^.next:=p;
  ELSE
    p^.next:=L^.current^.next;         (* p
                                 zwischenhängen *)
    p^.prev:=L^.current;
(*  L^.current^.next^.prev:=p; *)
    q:=L^.current^.next;  (* p als neuen
                          Vorgänger setzen *)
    q^.prev:=p;
    L^.current^.next:=p; (* p als neuen
                          Nachfolger setzen *)
  END;
  L^.current:=p;                          (*
                            current setzen *)
END AppendElement;

PROCEDURE RemoveElement(L:List);
VAR p:ListElementPointer;
BEGIN
  IF ~Empty(L) THEN
    p:=L^.current;  (* current merken *)
    IF (AtFirst(L) AND AtLast(L)) THEN
      L^.first:=NIL;   (* Leere Liste *)
      L^.last:=NIL;
      L^.current:=NIL;
    ELSIF AtFirst(L) THEN
      L^.first:=L^.current^.next;   (* first auf
                                      nächsten *)
      L^.first^.prev:=NIL;          (* prev des
                            neuen first auf NIL *)
```

→

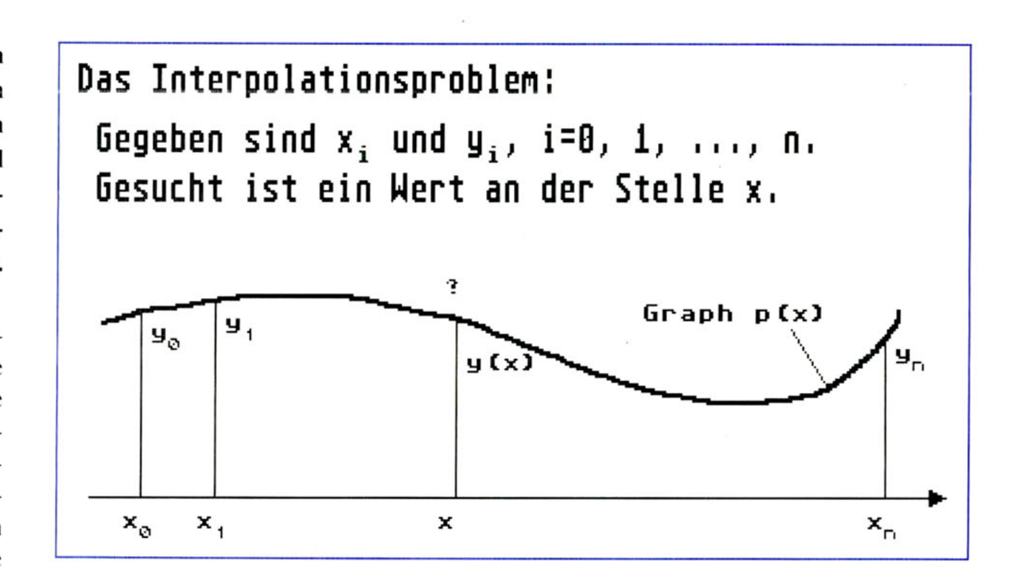
```
        L^.current:=L^.current^.next; (* current
                                        auf nächsten *)
      ELSIF AtLast(L) THEN
        L^.last:=L^.current^.prev;      (* last auf
                                           vorgänger *)
        L^.last^.next:=NIL;             (* next des
                                  neuen last auf NIL *)
        L^.current:=L^.current^.prev  (* current
                                      auf vorgänger *)
      ELSE
(*    L^.current^.prev^.next:=L^.current^.next;*)
        p^.prev^.next:=p^.next;         (* vorwärts
                                        über current *)
(*    L^.current^.next^.prev:=L^.current^.prev;*)
        p^.next^.prev:=p^.prev;          (* rückwärts
                                        über current *)
        L^.current:=L^.current^.next; (* current
                                        auf nächsten *)
      END;
      DEALLOCATE(p);                          (* altes
                                   current entfernen *)
    END;
  END RemoveElement;

END INTLists.
```

Numerische Mathematik

Teil 2
Interpolation

Nun sind wir schon bei der zweiten Folge. Heute befassen wir uns mit dem Interpolationsproblem. Bei diesem Problem geht man von einer Anzahl gegebener Wertepaare aus und bestimmt eine durch die gegebenen Punkte verlaufende, möglichst glatte Kurve.



Formulieren wir das Problem mathematischer: Gegeben sind n+1 Wertepaare *(xi,yi)* (i=0, 1, ..., n). Die *xi* sind paarweise verschieden. Gesucht ist nun eine Funktion p(x) derart, daß p*(xi)=yi* ist. Anschaulich ist dieses Problem in Bild 1 dargestellt. Da bekanntlich viele Wege nach Rom führen, gibt es auch hier mehrere Möglichkeiten, ein passende Funktion p(x) zu finden. Wir beschränken uns hier jedoch ausschließlich auf ganzrationale Funktionen p(x). p(x) ist also ein Polynom. Es gibt zwar auch andere Möglichkeiten, eine Interpolationsfunktion zu finden. Beispielsweise könnte p(x) auch gebrochen rational sein oder aus trigonometrischen Funktionen bestehen. Auch stückweise Interpolation z.B. mit Splines wollen wir hier nicht berücksichtigen. Ein Beispielprogramm zur Spline-Interpolation findet sich in [6] (s.u.).

p(x) heißt in unserem Fall das Interpolationspolynom, die *xi* heißen Stützstellen und die *yi* die zugehörigen Stützwerte. Bei n+1 gegebenen Stützstellen (mit den passenden Stützwerten) hat das Polynom p(x) höchstens den Grad n, d. h. es hat die Form *p(x)=a0+a1x+a2x2+...+anxn*, mit zu bestimmenden ai, i=0, 1, ..., n. Unter den genannten Voraussetzungen gibt es genau ein Interpolationspolynom. Unsere Aufgabe ist es nun, dieses zu finden. Beginnen wir mit der Interpolationsformel von Joseph Louis Lagrange (geb. 25. 1. 1738 in Turin, gest. 10. 4. 1813 in Paris). Die Lagrange'sche Interpolationsformel (Bild 2) besteht aus einer Summe mit den Stützwerten und Gewichten, in die die Stützstellen eingehen. Die Interpolationsformel löst die Interpolationsaufgabe, jedoch ist die Verwendung dieser Formel nicht gerade praktisch, da durch Hinzufügen weiterer Stützstellen die gesamte Formel neu berechnet werden müßte. Dafür ist sie jedoch von theoretischer Bedeutung, da man aus ihr z. B. Formeln zur Integration gewinnen kann. Näheres hierzu jedoch erst in Folge 3 dieser Serie.

$$p_n(x) = \sum_{i=0}^{n} y_i \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

$$p_n(x) = \frac{\sum_{i=0}^{n} \mu_i y_i}{\sum_{i=0}^{n} \mu_i} \quad \text{(baryzentrische Formel)}$$

$$\text{mit } \mu_i := \frac{\lambda_i}{x - x_i}, \; i = 0, 1, \ldots, n$$

$$\text{und } \lambda_i := \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{1}{x_i - x_j}$$

Bild 2: Lagrange'sche Interpolationsformel

Zur Auswertung des Interpolationspolynoms an anderen als den Stützstellen kann man die Rechnung etwas abkürzen. Die Formel, die zur Berechnung des Lagrange'schen Interpolationspolynoms herangezogen wird, heißt baryzentrische

Formel der Lagrange-Interpolation. Die *li* (l=Lambda, hier fehlt mir leider der griechische Buchstabe) werden rekursiv berechnet. Das Listing 1 ist dazu nur geringfügig zu verändern. Die Stützkoeffizienten *li* hängen nicht von der auszuwertenden Stelle ab, sie sind also nur einmal zu berechnen. Aber von dieser Stelle ab muß für jede neu auszuwertende Stelle x die Rechnung komplett neu durchgeführt werden. Somit hätten wir zwei Formen des Lagrange'schen Interpolationspolynoms besprochen.

$$p_n(x) = c_0 + c_1(x-x_0) + c_2(x-x_0)(x-x_1) + \ldots + c_n(x-x_0)(x-x_1)\ldots(x-x_{n-1})$$

mit $c_k := f[x_0, x_1, \ldots, x_k],\ k = 0, 1, \ldots, n$

und $f[x_i] = y_i,\ i = 0, 1, \ldots, n,$

$$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0},\quad f[x_1, x_2] = \frac{f[x_2] - f[x_1]}{x_2 - x_1},$$

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}, \ldots$$

und allgemein $f[x_{i_0}, x_{i_1}, \ldots, x_{i_k}] := \dfrac{f[x_{i_1}, x_{i_2}, \ldots, x_{i_k}] - f[x_{i_0}, x_{i_1}, \ldots, x_{i_{k-1}}]}{x_{i_k} - x_{i_0}}$

Bild 3: Newton'sche Interpolationsformel

Wenden wir uns einer anderen Interpolationsformel zu, der Interpolationsformel von Isaac Newton (geb. 4. 1. 1643 in Wollsthorpe (Lincolnshire), gest. 31. 3. 1727 in London). Wie in Bild 3 zu sehen ist, hat diese Formel einen ganz anderen Aufbau als die Formel von Lagrange. In den Koeffizienten der einzelnen Summanden tauchen nicht mehr jeweils alle *xi* auf, sondern sukzessive immer einer mehr als im Vorgänger. Dies ist in einem Programm besonders einfach auszuwerten. Das einzige Problem besteht in der Bestimmung der *ck*. Die *ck* werden über das Schema der sogenannte dividierten Differenzen berechnet. Das Schema erkennt man aus der im Bild 3 dargestellten Formel für die *ck* sowie dem Schema in Bild 4. Jedes *ck* baut auf zwei Formeln auf, die auf einer niedrigeren Stufe stehen. Auf der untersten Stufe werden die *f[xi]=yi* gesetzt. Besondere Vorteile bietet die Formel von Newton, weil man neue Stützstellen hinzufügen kann. Es ist pro hinzugefügter Stützstelle nur ein ck neu zu berechnen, welches man zudem noch teilweise aus alten Werten erhält. Aus den beschriebenen Interpolationsformeln von Lagrange und Newton erhält man leicht das Interpolationspolynom.

x_0 : $y_0 = f[x_0] = c_0$
$f[x_0,x_1] = c_1$
x_1 : $y_1 = f[x_1]$ — $f[x_0,x_1,x_2] = c_2$
$f[x_1,x_2]$ — $f[x_0,x_1,x_2,x_3] = c_3$
x_2 : $y_2 = f[x_2]$ — $f[x_1,x_2,x_3]$
$f[x_2,x_3]$
x_3 : $y_3 = f[x_3]$
...

Bild 4: Das Schema der dividierten Differenzen

Mit dem Algorithmus von Neville (Bild 5) wird die Aufgabe der Bestimmung eines zu interpolierenden Wertes direkt angegangen. Das Neville-Schema (Bild 6) ähnelt etwas dem Schema der dividierten Differenzen. Interpolation mit Polynomen hätten wir nun besprochen. Da mit der Anzahl der Stützstellen auch der Grad des Interpolationspolynoms steigt, kann man sich leicht überlegen, daß diese Art der Interpolation nicht für viele Stützstellen geeignet ist. Und genau so ist es auch. Man greift dann eher zu anderen Interpolationsmethode wie z. B. der Spline-Interpolation.

$$p_n(x) = p^*_{012\ldots n}(x)$$

mit $p^*_{i_0 i_1 \ldots i_k}(x) := \dfrac{(x - x_{i_0})\, p^*_{i_1 i_2 \ldots i_k}(x) - (x - x_{i_k})\, p^*_{i_0 i_1 \ldots i_{k-1}}(x)}{(x_{i_k} - x_{i_0})}$

und $p^*_k(x) = y_k,\ k = 0, 1, \ldots, n$

Bild 5: Neville'sche Interpolationsformel

Spline-Interpolation ist zwar auch Interpolation mit Polynomen, jedoch haben diese Polynome beispielsweise bei der kubischen Spline-Interpolation höchstens den Grad 3, unabhängig von der Anzahl der Stützstellen. Dadurch wird ein vernünftiger Verlauf der Interpolationsfunktion erreicht. Nehmen wir als Beispiel die kubische Spline-Funktion. Sie wird stückweise definiert. Zwischen zwei Stützstellen wird jeweils ein Polynom dritten Grades bestimmt. Der Verlauf der zusammengesetzten Graphen dieser Polynome ergibt dann den Verlauf des Graphen der Interpolationsfunktion. Hier merkt man schon den Unterschied zu den oben beschriebenen Interpolationsmethoden, bei nur ein Polynom gesucht wurde. Hier sind es bei n+1 Stützstellen bereits n Polynome. Eine kubische Spline-Funktion ist charakterisiert durch die gerade beschriebene Eigenschaft, auf jedem Teilintervall zwischen je zwei Stützstellen ein Polynom dritten Grades zu sein. Ferner ist die zweite Ableitung stetig (d. h. der zugehörige Graph läßt sich mit einem Stift zeichnen, ohne den Stift vom Papier abzuheben) an den inneren Stützstellen. Selbstverständlich erfüllt die Spline-Funktion auch die Forderungen des Interpolationsproblems. Diese Art der Interpolation wurde beispielsweise im Schiffsbau eingesetzt, um die Biegung einer Straklatte zu bestimmen. Straklatte sagt Ihnen nichts? Das englische Wort dafür ist Spline! Aber es gibt auch spielerische Anwendungen. Für einige Animationen und Computerspiele wird die Bahn eines fliegenden Objekts benötigt. Da diese Bahn schön rund aussehen soll, wird diese über Splines interpoliert. Bei der rationalen Interpolation wird eine gebrochenrationale Funktion R(x)=P(x)/Q(x) mit ganzrationalen Funktionen P und Q angesetzt. Diese Interpolationsfunktionen können nur in speziellen Fällen benutzt werden, da sie gegenüber den ganzrationalen Interpolationsfunktionen einen gewaltigen Nachteil haben: Sie erfüllen nicht in jedem Fall die Forderungen des Interpolationsproblems. Einige Punkte sind daher unerreichbar.

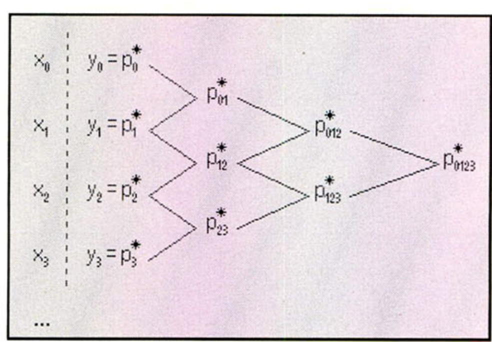


Bild 6: Das Schema von Neville

Im nächsten Monat werden wir teilweise aufbauend auf diese Folge die Themen Integralrechnung und Differentiation näher betrachten. Bis dahin...

Dipl.-Math. Dietmar Rabich

Ergebnisse der Programme

Wertetabelle, bestimmt mit Lagrange-Interpolation:		Wertetabelle, bestimmt mit Neville-Interpolation:		Wertetabelle, bestimmt mit Newton-Interpolation:	
x	p(x)	x	p(x)	x	p(x)
0.00000000	0.10000000	0.00000000	0.09999999	0.00000000	0.10000000
0.20000000	0.35348859	0.20000000	0.35348854	0.20000000	0.35348862
0.40000001	0.35791311	0.40000000	0.35791308	0.40000000	0.35791314
0.60000002	0.23721629	0.60000000	0.23721624	0.60000000	0.23721629
0.80000001	0.21728800	0.80000000	0.21728792	0.80000000	0.21728798
1.00000000	0.50000000	1.00000000	0.50000000	1.00000000	0.50000000
1.20000005	1.13724053	1.20000000	1.13724041	1.20000000	1.13724083
1.40000010	1.90494657	1.40000000	1.90494990	1.40000000	1.90495031
1.60000014	2.17714524	1.60000000	2.17715383	1.60000000	2.17715442
1.80000019	0.79999673	1.80000000	0.79999995	1.80000000	0.80000000

Literatur:

*[1] Einführung in die Numerische Mathematik I.
J. Stoer.
Springer Berlin/ Heidelberg/ New York/ Tokyo.
4. Aufl. 1983. S. 31ff*

*[2] Formelsammlung zur Numerischen Mathematik mit BASIC-Programmen.
G. Engeln-Müllges/ F. Reutter.
Bibliographisches Institut Mannheim/ Wien/ Zürich. 1. Aufl. 1983. S. 130ff*

*[3] Numerische Mathematik.
H. R. Schwarz.
Teubner Stuttgart. 1. Aufl. 1986. S. 88ff*

*[4] Numerische Methoden.
Å. Björck/ G. Dahlquist.
Oldenbourg München/ Wien. 2. Aufl. 1979. S. 201ff*

*[5] Methode der Numerischen Mathematik.
W. Böhm/ G. Gose/ J. Kahmann.
Vieweg Braunschweig/ Wiesbaden.
1. Aufl. 1985. S. 89ff*

*[6] Rund muß es sein.
D. Rabich.
ST Computer 1/89. S. 84ff*

*[7] Erfolgreich programmieren mit C.
J. A. Illik.
Sybex Düsseldorf/ San Francisco/ Paris/ London.
4. Aufl. 1987*

*[8] Programmieren in C.
B. W. Kernighan/ D. M. Ritchie.
Hanser München/ Wien. 1. Aufl. 1983*

*[9] PASCAL für Anfänger.
H. Schauer.
Oldenbourg Wien/ München. 4. Aufl. 1982*

*[10] PASCAL für Fortgeschrittene.
H. Schauer.
Oldenbourg Wien/ München. 2. Aufl. 1983*

*[11] Programmieren in Modula-2.
N. Wirth.
Springer Berlin/ Heidelberg/ New York/ Tokyo.
1. Aufl. 1985*


```
(**********************************************)
(* Beispielprogramm zur Lagrange-Interpolation.*)
(* --------------------------------------------*)
(* Entwickelt mit SPC Modula 2. 15.02.1989     *)
(**********************************************)

 (* --------- + ---------------- *)
 (* Listing 1 / (c) by D. Rabich *)
 (* --------- + ---------------- *)

 MODULE LagrangeInterpolation;

 (* Importe *)
 FROM InOut IMPORT WriteReal,WriteLn,WriteString,
                   Read;

 (* Konstanten *)
 CONST MaximalStuetz = 10;
       FloatFehler   = 1.0E-6;

 (* Variablen *)
 VAR Lambda,
     x,y        : ARRAY [0..MaximalStuetz] OF
                  REAL;
     xw         : REAL;
     MaxStuetz  : CARDINAL;
     VoidChar   : CHAR;

 (* Stützwerte einlesen *)
 PROCEDURE ReadStuetzStellen;

  BEGIN
   MaxStuetz:=5;
   x[0]:=-0.1;
   x[1]:= 0.0;
   x[2]:= 0.5;
   x[3]:= 0.7;
   x[4]:= 1.0;
   x[5]:= 1.8;
   y[0]:=-0.1;
   y[1]:= 0.1;
   y[2]:= 0.3;
   y[3]:= 0.2;
   y[4]:= 0.5;
   y[5]:= 0.8
  END ReadStuetzStellen;

 (* Lambda-Koeffizienten berechnen *)
 PROCEDURE CalcLambda;

  VAR i,k   : CARDINAL;
      Summe : REAL;

  BEGIN
   Lambda[0]:=1.0;
   FOR k:=1 TO MaxStuetz DO
    FOR i:=0 TO k-1 DO
     Lambda[i]:=Lambda[i]/(x[i]-x[k])
    END;
    Summe:=0.0;
    FOR i:=0 TO k-1 DO
     Summe:=Summe+Lambda[i]
    END;
    Lambda[k]:=-Summe
   END
  END CalcLambda;

 (* Mu-Koeffizienten berechnen *)
 PROCEDURE CalcMuSums (BerX : REAL; VAR Sum1,Sum2
                            : REAL);
```

```

  VAR i  : CARDINAL;
      Mu : REAL;

  BEGIN
   Sum1:=0.0;
   Sum2:=0.0;
   FOR i:=0 TO MaxStuetz DO
    Mu  :=Lambda[i]/(BerX-x[i]);
    Sum1:=Sum1+Mu*y[i];
    Sum2:=Sum2+Mu
   END
  END CalcMuSums;

 (* Interpolationswert berechnen *)
 PROCEDURE Interpol (BerX : REAL) : REAL;

  VAR i          : CARDINAL;
      Sum1,Sum2  : REAL;

  BEGIN
   ReadStuetzStellen;

   (* Ist BerX Stützstelle? *)
   FOR i:=0 TO MaxStuetz DO
    IF ABS(x[i]-BerX)<FloatFehler THEN
     RETURN y[i]
     END
   END;

   (* Wert berechnen *)
   CalcLambda;
   CalcMuSums(BerX,Sum1,Sum2);
   RETURN Sum1/Sum2

  END Interpol;

 (* Hauptprogramm *)
 BEGIN

  xw:=0.0;
  WriteString('    x              p(x)');
  WriteLn;

  REPEAT
   WriteReal(xw,12,8);
   WriteString(' ');
   WriteReal(Interpol(xw),12,8);
   WriteLn;
   xw:=xw+0.2
  UNTIL xw>2.0;

  Read(VoidChar)
 END LagrangeInterpolation.
```

```
(**********************************************)
(* Beispielprogramm zur Newton-Interpolation.  *)
(* ------------------------------------------*)
 (* Entwickelt mit ST Pascal Plus.  13.02.1989 *)
 (**********************************************)

 (* --------- + ---------------- *)
 (* Listing 2 / (c) by D. Rabich *)
 (* --------- + ---------------- *)

 program newton_interpolation;

 (* Konstante *)
 const max_dim = 10;

 (* Variablen *)
 var max_stuetz : short_integer;
     x,y,
     c          : array [0..max_dim] of real;
     xw         : real;

 (* Werte einlesen *)
 procedure read_werte;

  begin
   max_stuetz:=5;
   x[0]:=-0.1;
   x[1]:= 0.0;
   x[2]:= 0.5;
   x[3]:= 0.7;
   x[4]:= 1.0;
   x[5]:= 1.8;
   y[0]:=-0.1;
   y[1]:= 0.1;
   y[2]:= 0.3;
   y[3]:= 0.2;
   y[4]:= 0.5;
   y[5]:= 0.8
  end;

 (* Koeffizienten berechnen *)
 procedure calc_coeff;

  var i,k : short_integer;

  begin
   for i:=0 to max_stuetz do
    c[i]:=y[i];
   for k:=1 to max_stuetz do
    for i:=max_stuetz downto k do
     c[i]:=(c[i]-c[i-1])/(x[i]-x[i-k])
  end;

 (* Wert interpolieren *)
 function interpol (ber_x : real) : real;

  var i : short_integer;
      p : real;

  begin
   read_werte;
   calc_coeff;

   (* Polynom auswerten *)
   p:=c[max_stuetz];
   for i:=max_stuetz-1 downto 0 do
    p:=c[i]+(ber_x-x[i])*p;

   interpol:=p
  end;

 (* Hauptprogramm *)
 begin
  xw:=0.0;
  writeln('    x              p(x)');

  (* Werte ausgeben *)
  repeat
   writeln(xw:12:8,' ',interpol(xw):12:8);
   xw:=xw+0.2
  until xw>2.0;

  (* auf Tasten warten... *)
  repeat
  until keypress

 end.
```

```
/**********************************************/
/* Beispielprogramm zur Neville-Interpolation. */
/* ------------------------------------------ */
/* Entwickelt mit Turbo C.          13.02.1989 */
 /**********************************************/

 /* --------- + ---------------- */
 /* Listing 3 / (c) by D. Rabich */
 /* --------- + ---------------- */

 /* Ein-/Ausgabe-Routinen importieren */
 # include <stdio.h>

 /* Konstante */
 # define MAXDIM 10

 /* globale Variablen */
 int   max_stuetz;
```

→

```
float x[MAXDIM],y[MAXDIM],p[MAXDIM];

/* Werte einlesen */
void read_werte (void)

 {
  max_stuetz=5;
  x[0]=-0.1;
  x[1]= 0.0;
  x[2]= 0.5;
  x[3]= 0.7;
  x[4]= 1.0;
  x[5]= 1.8;
  y[0]=-0.1;
  y[1]= 0.1;
  y[2]= 0.3;
  y[3]= 0.2;
  y[4]= 0.5;
  y[5]= 0.8;
 }

/* Wert interpolieren */
float interpol (float ber_x)

 {
  int i,k;

  read_werte();

  for(i=0;i<=max_stuetz;i++)
   p[i]=y[i];
  for(k=1;k<=max_stuetz;k++)
   for(i=max_stuetz;i>=k;i--)
    p[i]+=(ber_x-x[i])*(p[i]-p[i-1])/(x[i]-x[i-k]);

  return(p[max_stuetz]);
 }

/* Hauptprogramm */
void main (void)

 {
  float xw;

  printf("    x              p(x)\n");

  for(xw=0.0;xw<=2.0;xw+=0.2)
   printf("%12.8g %12.8g\n",xw,interpol(xw));

  getchar();
 }
```

Lovely Helper

Ein Desk-Accessory 6

Der Directorydruck

Bereits nach einer sehr kurzen Bekanntschaft mit meinem ST war mir klar - eins gefällt mir an dieser Kiste überhaupt nicht: Die unumgängliche Unordnung, die sich bereits bei dem Besitz von mehr als zwei Disketten in jeden Dateibestand einschleicht, läßt sich leider nur sehr unzureichend dokumentieren. So ist es, zur Dokumentation einer Diskette, die über mehrere Unterverzeichnisse verfügt, durchaus nicht befriedigend, eine Hardcopy jedes Unterverzeichnisses anlegen zu müssen, um ein wenig Übersicht über das Ganze zu erhalten. Mit der Zeit - und der Erwerbung einer Harddisk - stieg diese Unzufriedenheit ganz gewaltig, weshalb ich mich genötigt sah, ein entsprechendes Utility zu formulieren. Um im Dschungel der mittlerweile ungezählten Accessories gleicher Thematik nicht völlig unterzugehen, hält dieses Utility, eingebaut in unseren Lovely Helper, einige Leckerbissen für Sie bereit:

1. Zusätzlich zum Ausdruck des Hauptverzeichnisses kann optional der Ausdruck aller Unterverzeichnisse geschehen.

2. Die Sortierung erfolgt nach denselben vier Kriterien wie im Betriebssystem, also nach Name, Größe, Datum oder Typ.

3. Der Ausdruck geschieht über unsere Spoolerroutinen. Dadurch werden sämtliche, eventuell im Inhaltsverzeichnis enthaltene Sonderzeichen - etwa Umlaute - auch auf Ihrem Drucker korrekt wiedergegeben.

Die XENIX-Struktur des TOS

Punkt 1 - der Ausdruck aller Unterverzeichnisse - enthält dabei den interessantesten Algorithmus. Zu seinem Verständnis möchte ich etwas weiter ausholen. Stichwort ist die XENIX-Struktur des TOS. Gemeint ist damit die "Art und Weise", wie unter TOS Dateien verwaltet werden, also Strukturierung von Dateien durch Unterverzeichnisse, die hierarchische Schachtelung derselben etc. Wie Sie vielleicht wissen, ist diese Struktur keine ATARI-spezifische Entwicklung, sondern fand bereits in vielen anderen erfolgreichen Betriebssystemen, wie etwa MS-DOS oder UNIX, Anwendung. Der Ursprung ist dabei ein XENIX genanntes Betriebssystem, wodurch sich die Bezeichnung erklärt. In einem derartigen Betriebssystem kann man sich nun alle Dateien einer Station in einer baumartigen Struktur organisiert denken (Abbildung 21). Die Wurzel des Baumes ist dabei das Hauptverzeichnis der Station und wird mit dem Zeichen '\' benannt. Anmerkung: Komischerweise stehen Bäume in der Informatik immer auf dem Kopf. Ich schließe mich dieser Konvention an. Wenn Sie die Wurzel suchen: Sie ist oben.

Das Hauptverzeichnis kann nun sowohl Dateien, als auch Unterverzeichnisse, im ST-Jargon auch Ordner genannt, enthalten. Mit den Ordnern verhält es sich genauso. Auch sie dürfen sowohl Dateien, als auch ihrerseits wieder Ordner enthalten. Auf einzelne Elemente des Dateibaumes, egal ob es sich dabei um Ordner oder Dateien handelt, greift man über die zugehörigen Pfadnamen zu. Vergleichen Sie dazu bitte auch Abbildung 21. Dargestellt ist ein Dateibaum mit sämtlichen zugehörigen Pfadnamen. Da diese Pfadnamen bei größeren Baumstrukturen, wie etwa auf einer Hard-Disk, mitunter recht lang werden, führt man, zwecks einfacherer Handhabung, noch den Begriff des aktuellen Ordners ein. Dieser Ordner ist unter sämtlichen Ordnern des Dateibaumes frei wählbar und kann jederzeit geändert werden. Zum Zugriff auf Dateien innerhalb des aktuellen Ordners braucht man nicht mehr den kompletten Pfadnamen anzugegeben. Es genügt der einfache Dateiname. Auch Dateien "in der Nähe" des aktuellen Ordners lassen sich einfacher beschreiben als durch ihren kompletten Pfad. Gehen wir z.B. davon aus, daß wir uns im Ordner *\TEIL.6\RSC* innerhalb der Abbildung 21 befinden. Untenstehende Tabelle enthält dann jeweils zwei Dateinamen, die ein und dieselbe Datei innerhalb der Baumstruktur beschreiben.

\TEIL.6\RSC\DIRECT.I	DIRECT.I
\TEIL.6	..
\TEIL.6\RUN	..\RUN
\TEIL.6\SOURCE\DIRECT.I	..\SOURCE\DIRECT.I
\TEIL.5	..\..\TEIL.5
\	..\..

```
\TEIL.5
\TEIL.6
\TEIL.6\ARTIKEL
\TEIL.6\ARTIKEL\ART_06.DOC
\TEIL.6\ARTIKEL\ABB_20.PIC
\TEIL.6\ARTIKEL\ABB_21.PIC
\TEIL.6\ARTIKEL\ABB_22.PIC
\TEIL.6\ARTIKEL\ABB_23.TXT
\TEIL.6\RSC
\TEIL.6\RSC\DIRECT.I
\TEIL.6\RSC\DIRECT.RSC
\TEIL.6\RSC\DIRECT.RSD
\TEIL.6\RUN
\TEIL.6\RUN\DIRECT.ACC
\TEIL.6\RUN\DIRECT.RSC
\TEIL.6\SOURCE
\TEIL.6\SOURCE\DIRECT.I
\TEIL.6\SOURCE\DIRECT.PAS
\TEIL.6\SOURCE\DIRECT1.PAS
\TEIL.7
```

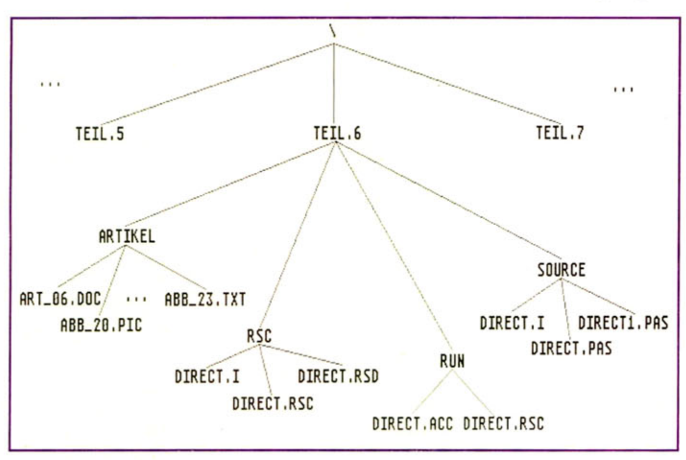


Abb. 21: Der Aufbau des Directoys gleicht einem Baum

Die Verwendung der beiden Punkte innerhalb der Pfadnamen bedeutet dabei soviel wie "Gehe einen Ordner im Dateibaum zurück!". Damit ist es relativ bequem, auch an den äußeren Enden des Dateibaumes Dateien in unmittelbarer Nähe anzusprechen, ohne immer über den ganzen Pfad steigen zu müssen. Für unsere heutige Anwendung benötigen wir nun einen Algorithmus, der in einer "gewissen" Reihenfolge nacheinander jeweils alle Ordner eines Dateibaumes zu aktuellen Ordnern macht, um dann den jeweiligen Ordner auszudrucken. Die Arbeit läßt sich leicht in zwei Teile zerlegen:

1. Der Ausdruck des aktuellen Ordners

2. Der Durchlauf durch sämtliche Ordner des Dateibaumes, mit jeweiligem Aufruf von Punkt 1.

Ausdruck eines Ordners

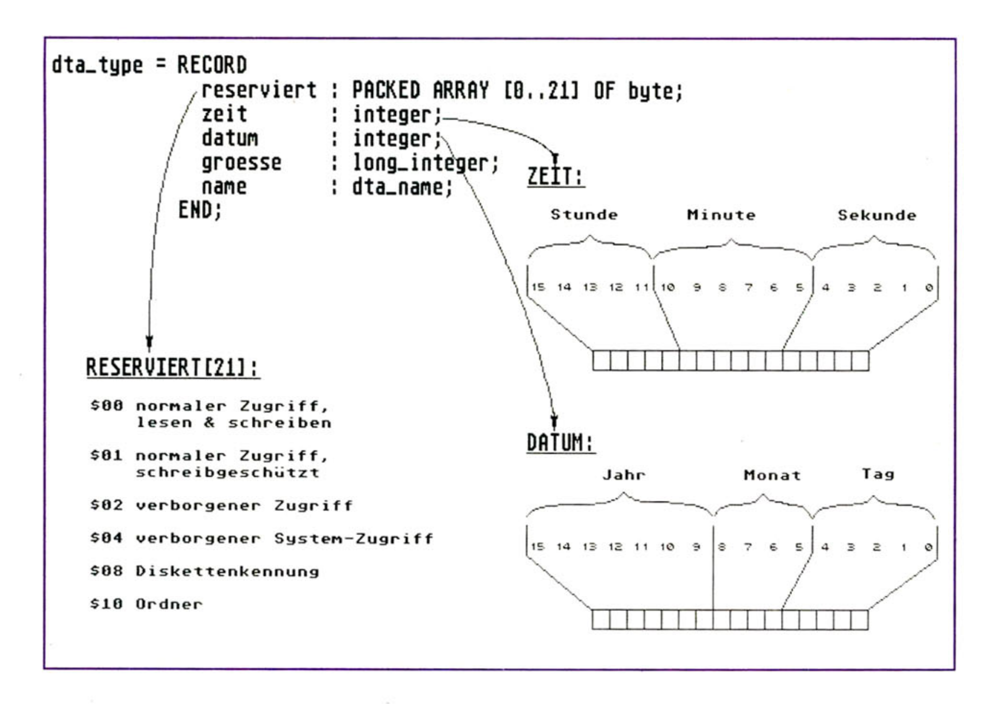
Beginnen wir mit Punkt 1. Dazu ist es zunächst erforderlich, irgendwie an die Informationen, sprich Dateieinträge, des aktuellen Ordners heranzukommen. TOS stellt für diese Aufgabe die **D**isk-**T**ransfer-**A**dress, kurz DTA, zur Verfügung. An diese Adresse schreibt TOS einen 44 Bytes großen Informationsblock, der sämtliche notwendigen Daten über eine Datei enthält. Das Format dieses DTA-Puffer können Sie der Abbildung 20 entnehmen (Innerhalb unserer Dateien, wurde der DTA-Puffer bereits in der Datei *HILF.PAS* - in der ersten Folge des Lovely Helper - deklariert.). Erster Bestandteil des Puffers ist ein Array aus 22 Bytes. Es enthält einige Organisations-Informationen für das Betriebssystem. Für uns wichtig ist nur das letzte Element dieses Arrays - *reserviert[21]*. Es enthält ein Bitmuster, das Aufschluß über den Dateityp der betrachteten Datei gibt. Alle sechs möglichen Belegungen dieses Bytes können Sie der Abbildung 20 entnehmen. Die nächsten beiden Informationen beziehen sich auf das Erstell- bzw. Modifikationsdatum der betrachteten Datei. Die beiden Variablen *zeit* und *datum* nehmen die entsprechende Information auf.

Da zur Speicherung beider Größen hier aber nur zwei Integer-Werte (2 x 16 Bit) verwendet werden, wird eine gewisse Interpretation der Größen notwendig. Wie Sie ebenfalls der Abbildung entnehmen können, sind die sechs Größen Tag, Monat, Jahr, Stunde, Minute und Sekunde auf folgende Weise gepackt:

Variable *zeit*:		Bits
	Stunde	11-15
	Minute	5-10
	Sekunde	0-4
Variable *datum*:		**Bits**
	Jahr	9-15
	Monat	5-8
	Tag	0-4

Einer Erklärung bedürfen dabei die Werte für das Jahr und die Sekunde. Hier reicht nämlich der darstellende Bereich nicht für die Aufnahme sämtlicher möglicher Zustände aus. Der Bereich für Sekunde kann maximal 32 (2^5) Zustände annehmen, wodurch es nur möglich ist, jede zweite Sekunde zu zählen. Zu den Jahren ist immer der Offset 1980 hinzuzuaddieren, um die korrekte Jahreszahl zu erhalten. Auf diese Weise kann unsere ST Systemuhr maximal bis zum Jahre 2107 (1980 + 2^7 - 1) Daten korrekt darstellen. Wohl mehr als ausreichend, selbst für den bis jetzt ziemlich zähen und langlebigen ST. Vierter Bestandteil des DTA-Puffers ist der Long_integer *groesse*. Er enthält die Länge der betrachteten Datei in der Einheit Byte. Letzter Bestandteil, der String *name* (dta_name = PACKED ARRAY [1..14] OF char), enthält den Namen der Datei. Er wird, wie im Betriebssystem üblich, mit *CHR$(0)* abgeschlossen. Die Adresse des DTA-Puffers verrät uns TOS mit dem Betriebssystemaufruf fgetdta (GEMDOS($2f)). Anmerkung: In der Regel liegt die DTA außerhalb des Bereichs der Pascal-Variablen. Deshalb ist *p-* als Compileroption unverzichtbar, sobald mit fgetdta gearbeitet wird. Bleibt die Frage, mit welchen Befehlen wir Daten in diesen DTA-Puffer bekommen. Dazu stellt TOS das Operationspaar *fsfirst* (GEMDOS($4e)) und *fsnext* (GEMDOS($4f)) zur Verfügung. *fsfirst* erhält als ersten Parameter einen Pfadnamen, im Betriebssystemformat, also *cstring*. Er darf auch Wildcards ('*' und '?') beinhalten. Als weiterer Parameter wird ein Integer übergeben, der die Art der gesuchten Datei beschreibt. Das Format entspricht dabei dem des 22. Byte des DTA-Puffers *(reserviert[21])*.

Zusätzlich können hier jedoch Kombinationen der sechs möglichen Zustände angegeben werden, indem man die entsprechenden Werte "verodert". So entspricht ein Wert von $17 (folder | disk_label | hidden_sys | hidden | read_only | normal_file) für diesen Parameter, der Suche nach einem beliebigen Ordnereintrag. Berechnet wird von *fsfirst* ein Bool'scher Parameter, entsprechend dem Erfolg der durchgeführten Suche. War die Suche mit *fsfirst* erfolgreich, so stehen die genauen Informationen der aufgefundenen Datei automatisch im DTA-Puffer. Mit *fsnext* kann dann nach weiteren, zutreffenden Einträgen gesucht werden. Wird für *fsfirst* als Dateiname '*.*' angegeben und als Attribut $11 (normal_file | read_only | folder), so kann mit den nachfolgenden *fsnext*-Aufrufen ein komplettes Inhaltsverzeichnis erfragt



werden. Dabei werden alle Dateien gefunden, die auch in den GEM-Windows angezeigt würden. Im Prinzip also sehr einfach, nämlich:

```
IF fsfirst( ... ) THEN
  BEGIN
   ausgabe( ....);
   WHILE fsnext DO
    ausgabe( ... );
  END;
```

Umgesetzt in die Praxis benötigen wir nun jedoch noch einen ganzen Satz von Operationen, um eine maßgeschneiderte Ausgabe eines Ordners vorzunehmen (Listing 12, Zeilen 10-276). Die ersten beiden Prozeduren - *inter_date* und *inter_time* (Zeilen 10-30) - wandeln einen Integerwert, in einem der zwei Systemzeitformate, in jeweils drei Integerwerte, in normalem Format, um. Die Prozedur *ausgabe_dta* (Zeilen 32-82) erhält eine Variable *dta* im Format des DTA-Puffers. Sie druckt eine Zeile mit den entsprechenden Informationen aus. Benutzt werden dazu unsere Spoolerroutinen *send_signal* und *send_string*, sowie einige Routinen zum Formatieren der beteiligten Variablen. Die eigentliche Arbeit aber erledigt Prozedur *directory* (Zeilen 99-276). Wir wenden uns zunächst ihrem Anweisungsteil (Zeilen 240-276) zu. Im ersten Teil (Zeilen 241-249) wird die DTA-Adresse und der Name des aktuellen Ordners ermittelt. Der Name des Ordners wird daraufhin ausgegeben. In den Zeilen 250-261 wird nun - mit dem Dateinamen '*.*' - das komplette Ordnerverzeichnis zunächst von der Station gelesen und in einem Puffer *(dta_puffer)* abgelegt. Grund für dieses Vorgehen ist, daß wir *fsfirst* keine Sortierkriterien für die gesuchten Dateien mitgeben können. Wir haben also alle Informationen zuerst zu puffern und dann selber zu sortieren, bevor an die Ausgabe gedacht werden kann. Die Sortierung (Zeile 264) erfolgt mit dem bereits früher vorgestellten Quicksort (Zeilen 205-238). Es ist diesmal lediglich unserer Problematik angepaßt worden. Größte Schwierigkeit war dabei die Wahl einer zutreffenden Relation zur Anordnung der Dateien. Relation *rel* (Zeilen 115-203) leistet das Gewünschte. Hier wird zunächst berücksichtigt, welcher Art das Sortierkriterium ist. Zur Verfügung stehen Name. Zeit. Größe und Typ. Daraufhin erfolgt die prompte Auswertung der entsprechenden (Teil-)Relation. Nach diesem Sortiervorgang wird nun nur noch der Puffer, durch mehrfachen Aufruf von *ausgabe_dta*, ausgegeben.

Ein wenig Rekursion x_directory

Kommen wir nun zu der Prozedur, die ein komplettes Stationsverzeichnis erstellt - *x_directory* (Zeilen 278-348). Bis auf die Ausgabe der eingeschachtelten Ordner, sowie ein wenig Randinformation - Stationskennung und Stationsbelegungsgrad ermitteln und ausgeben -, hat *x_directory* nur einen Durchlauf durch den Dateibaum durchzuführen. Dies geschieht in der rekursiven Prozedur *dfs*, Zeilen 285-311. Benutzt wird hier im Wesentlichen der Betriebssystemaufruf *dsetpath* (GEMDOS($3b)) im Zusammenwirken mit *fsfirst* und *fsnext*. Begonnen wird der Durchlauf im Hauptverzeichnis '\' (Die Initialisierung dazu erfolgt noch in *x_directory* selber (Zeilen 315-317)). *dfs* gibt nun zunächst das komplette Hauptverzeichnis aus und ermittelt dann mit fsfirst und fsnext sämtliche verzeichneten Ordner im aktuellen Verzeichnis. Dabei erfolgt jeweils ein rekursiver Aufruf mit dem entsprechenden Ordnernamen, sowie ein Rücksprung mit dem Pfadnamen '..' (Ordner zurück) nach erfolgter Rekursion. Besondere Bedeutung kommt dabei den reservierten Bytes des DTA-Puffers zu. In Ihnen wird nämlich vermerkt, inwieweit mit *fsfirst* und *fsnext* ein Ordner bereits durchwandert ist. Durch die rekursiven Aufrufe - und damit andere Belegungen des DTA-Puffers - wird diese Information überschrieben. Beim Durchlauf würde dies bei der Unterlassung von Gegenmaßnahmen dazu führen, daß immer wieder in den ersten Ordner hineingestiegen wird: Eine Endlosschleife, über die sich besonders der Drucker freut. Eine einfache Gegenmaßnahme, die keinerlei Information über die genaue Struktur der reservierten Bytes erfordert, ist die komplette Sicherung des DTA-Puffers in einer Variablen *copy* (Zeile 303). Nach dem rekursiven Aufruf, kann dann auf sehr einfache Weise - Zuweisung von

```
PROCEDURE where(compare : string);

  VAR pfad     : string;
      cpfad    : cstring;
      name     : string;
      dta_ptr  : dta_ptr_type;

  BEGIN
    dta_ptr:=fgetdta;
    IF dgetpath(cpfad,0) THEN
      BEGIN
        ctopstr(cpfad,pfad);
        ptocstr(compare,cpfad);
        IF fsfirst(cpfad,normal_file | read_only |
              folder)>=0           THEN
          BEGIN
            dtopstr(dta_ptr^.name,name);
            writeln(concat(pfad,'\',name));
            WHILE fsnext DO
              BEGIN
                dtopstr(dta_ptr^.name,name);
                writeln(concat(pfad,'\',name);
              END;
          END;
      END;
  END;
```

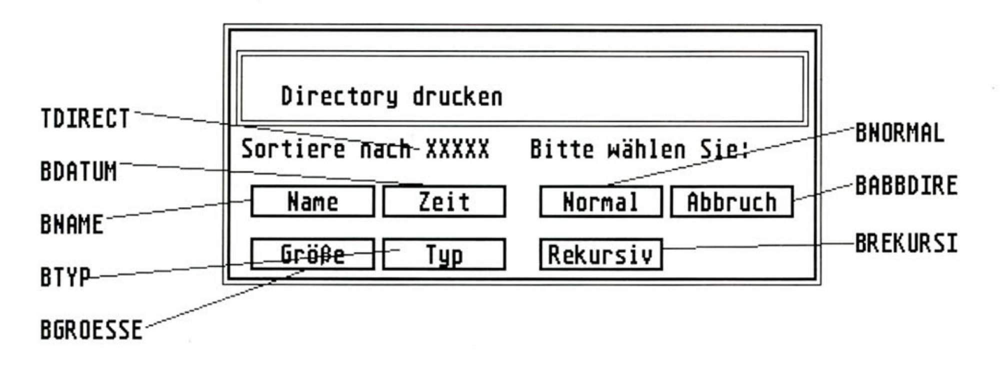


Abb. 22

copy an den DTA-Puffer - der ursprüngliche Zustand wieder hergestellt werden. Für Insider: Das, was wir gerade gemacht haben, ist an sich nichts weiter, als ein Preorder-Durchlauf durch den Dateibaum, mit der Knotenoperation "directory". Denkbar sind durchaus auch andere Knotenoperationen: Ersetzen wir etwa *directory*, sowie Ihre Aufrufe durch die Operation, die im Listing auf der vorigen Seite zu sehen ist, so erhalten wir - wird das Ganze mit *dfs* in ein separates Programm eingebunden - eine Routine, die in größeren Dateibeständen bestimmte Dateien sucht. Enthält *compare* beispielsweise den Wert '*.PAS', so werden sämtliche Pascal-Dateien mit ihrem kompletten Pfadnamen ausgegeben. Zumindestens in UNIX, mittlerweile aber auch auf dem ST, habe ich das immer sehr nützlich gefunden.

Objekt:	Objektart:	Länge:	Diverses:
TDIRECT	TEXT	5	
BNAME	BUTTON		Flags: Selectable & Exit
BGROESSE	BUTTON		Flags: Selectable & Exit
BDATUM	BUTTON		Flags: Selectable & Exit
BTYP	BUTTON		Flags: Selectable & Exit
BNORMAL	BUTTON		Flags: Selectable & Exit
BABBDIRE	BUTTON		Flags: Selectable & Exit
BREKURSI	BUTTON		Flags: Selectable & Exit

Tab. 1: Objekte der Abb. 22

Wieder ein wenig GEM

Nach diesem Ausflug in die Welt der Betriebssysteme, ist nun noch das Bindeglied zwischen den bis jetzt vorgestellten Operationen und unserem Lovely Helper zu formulieren. Benutzt wird der Dialog *DIRECT*, Abbildung 22, Tabelle 1.

Der einzige Text dieses Dialoges - *TDIRECT* - dient dabei dazu, daß aktuelle Sortierkriterium anzuzeigen. Die vier darunter befindlichen Feldtasten dienen der Selektion dieses Kriteriums. Die Taste *BNORMAL* ermöglicht den Ausdruck eines normalen (nicht rekursiven) Hauptverzeichnisses. Entsprechend leitet *BREKURSI* die rekursive Variante ein. *BABBDIRE* letztendlich dient dem Abbruch des *DIRECT*-Dialoges. Anmerkung: Beachten Sie bitte auch, daß für das heutige Resource - *DIRECT.RSC* - wieder die beiden Dialoge *PARAMETE* und *SYNCHRO* benötigt werden, damit die Druckerparameter eingestellt werden können. Die Dialogverwaltung (*do_direct*, Zeilen 350-391) ist, wie schon der Dialog, sehr schlicht. In der *REPEAT*-Schleife, die die Dialogausführung einschließt (Zeilen 360-377), wird vor *do_dialog* der Text *TDIRECT* initialisiert.

Nach *do_dialog* wird zwischen den vier Feldtasten, zur Wahl des Sortierkriteriums, und den restlichen drei Feldtasten unterschieden. Bei Wahl einer der vier Tasten für die Sortierkriterien, wird nur der entsprechende Parameter (*default_sort*) umgesetzt und der Dialog wiederholt. Bei Wahl einer anderen Taste wird die Schleife verlassen; es erfolgt eine abschließende Auswertung. In dieser Auswertung ist zunächst, wie schon in der letzten Folge, zu prüfen, ob der Drucker nicht vielleicht bereits für unseren Spooler arbeitet (spoolstatus = unused). Ist diese Bedingung erfüllt - wir stören den Spooler nicht -, so ist nur noch zwischen einem rekursiven Inhaltsverzeichnis (*BREKURSI*) und einem einfachen Hauptverzeichnis (*BNORMAL*) zu unterscheiden und in die entsprechende Prozedur zu verzweigen.

Das Accessory DIRECT

Die Accessory-Verwaltung von *DIRECT.ACC* (Listing 13) hält nun nichts Aufregendes mehr für uns bereit. Die Möglichkeit mehrere Menüleisten von einem Accessoir aus zu belegen, hatten wir bereits beim letzten Mal kennengelernt. Ach ja: Die Definition von bundesland in den Zeilen 19-21. Ein kleiner Pfusch! Da die Parameterverwaltung auch das *Bundesland* für unser *ZEIT.ACC*

Erweiterte Directory des Massenspeichers A

Label des Speichermediums :

EP.14B	0	00-00-1980	00:00

Directory :

O TEIL.5	0	18-06-1988	17:24

Directory : \TEIL.5

O ARTIKEL	0	18-06-1988	17:24
O RSC	0	18-06-1988	17:26
O RUN	0	18-06-1988	17:26
O SOURCE	0	18-06-1988	17:26

Directory : \TEIL.5\ARTIKEL

ART_05.DOC	16013	18-06-1988	17:24
ABB_15.PIC	32000	18-06-1988	17:24
ABB_16.PIC	32000	18-06-1988	17:25
ABB_17.PIC	32000	18-06-1988	17:25
ABB_18.PIC	32000	18-06-1988	17:25
ABB_19.TXT	3948	18-06-1988	17:25

Directory : \TEIL.5\RSC

ZEIT.I	5295	18-06-1988	17:26
ZEIT.RSC	8998	18-06-1988	17:26
ZEIT.RSD	1872	18-06-1988	17:26

Directory : \TEIL.5\RUN

ZEIT.ACC	43055	18-06-1988	17:26
ZEIT.RSC	8998	18-06-1988	17:26

Directory : \TEIL.5\SOURCE

ZEIT.I	5295	18-06-1988	17:27
ZEIT.PAS	3129	18-06-1988	17:27
ZEIT2.PAS	14451	18-06-1988	17:27

Bytes gesamt	:	728064
Bytes frei	:	473088
Bytes belegt	:	254976

Abb. 23: Ein rekursives Inhaltsverzeichnis

zu verwalten hat, ist diese Deklaration in Listing 13 aufzunehmen, um die Datei *SPOOLER1.PAS* unverändert einbinden zu können. So! Für heute bin ich damit wieder einmal am Ende (meiner Ausführungen). Als letztes möchte ich Ihnen noch Abbildung 23 presentieren: Das Resultat unserer heutigen Bemühungen, angewendet auf eine meiner Datendisketten zum Lovely Helper.

Vorausschau

Beim nächsten Mal wird Sie ein naturwissenschaftlicher Taschenrechner erwarten, der es schon ein bißchen in sich hat. Er verfügt über:

- Punkt-vor-Strich-Rechnung
- Klammerung (die zulässigen Klammerebenen können Sie selbst bestimmen)
- Die wichtigsten naturwissenschaftlichen Funktionen (sin, cos, tan, ln, ...) und ihre Umkehrungen
- Die trigonometrischen Funktionen können in allen, geläufigen Einheiten (DEG, RAD und GRAD) durchgeführt werden

Mal wieder was ganz und gar Theoretisches und nicht wie heute aus dem Sumpf der Betriebssystemprogrammierung. In der Hoffnung, daß Ihnen diese Wechselkost gut bekommt, verbleibe ich bis dahin.

Dirk Brockhaus

```
{***********************************************}
{* Listing 13 : Resource-Handling für die      *}
{*              Directoryausgabe               *}
{*                                             *}
{* Datei       : DIRECT.PAS                    *}
{* last update: 27.5.88                        *}
{***********************************************}

 {$s20,p-}

 PROGRAM directory;

 CONST {$i gemconst.pas}
       {$i trixcons.pas}
       {$i direct.i}

 TYPE {$i gemtype.pas}
      {$i trixtype.pas}

      bundesland = (schleswig_holstein,hamburg,
                    bremen,niedersachsen,
                    nordrhein_westfalen,hessen,
                    rheinland_pfalz,
                    saarland,baden_wuertemberg,
                    bayern);

 VAR msg              : message_buffer;
     apl_name_1       ,
     apl_name_2       : str255;
     apl_nr           ,
     menu_nr_1        ,
     menu_nr_2        ,
     event            ,
     dummy            : short_integer;

     direct_dialog    ,
     parameter_dialog ,
     synchro_dialog   : dialog_ptr;

 {$i gemsubs.pas}
 {$i trixsubs.pas}
 {$i hilf.pas}
 {$i spooler1.pas}
 {$i direct1.pas}

 FUNCTION initialisieren : boolean;

   VAR ok : boolean;

   BEGIN
     ok:=load_resource('A:\DIRECT.RSC');
     IF ok THEN
       BEGIN
         apl_name_1:='  Directory';
         menu_nr_1:=menu_register(apl_nr,apl_name_1
                                  );
         apl_name_2:='  Druckerparameter';
         menu_nr_2:=menu_register(apl_nr,apl_name_2
                                  );
         find_dialog(paramete,parameter_dialog);
         find_dialog(synchro,synchro_dialog);
         find_dialog(direct,direct_dialog);
         center_dialog(parameter_dialog);
         center_dialog(synchro_dialog);
         center_dialog(direct_dialog);
         io_check(false);
         rewrite(spoolchannel,'PRN:');
         load_parameter;
       END;
     initialisieren:=ok;
   END;

 BEGIN
   apl_nr:=init_gem;
   IF apl_nr>=0 THEN
     IF initialisieren THEN
       WHILE true DO
         BEGIN
           event:=get_event(e_message,0,0,0,0,false
                            ,0,0,0,0,
                            false,0,0,0,0,msg,dummy
                            ,dummy,dummy,
                            dummy,dummy,dummy);
           IF (event & e_message<>0) AND (msg[0]=
                                       ac_open) THEN
             BEGIN
               IF msg[4]=menu_nr_1 THEN
                 do_direct;
               IF msg[4]=menu_nr_2 THEN
                 do_parameter;
             END;
         EnD;
 END.
```

```
{***********************************************}
{* Listing 12 : Routinen zum Ausdruck eines    *}
{*              normalen und eines             *}
{*              rekursiven Partitions-         *}
{*              Inhaltsverzeichnisses          *}
{*              und das zugehörige             *}
{*              Dialog-Handling                *}
{*                                             *}
{* Datei       : DIRECT1.PAS                   *}
{* last update: 24.5.1988                      *}
{***********************************************}

PROCEDURE inter_date(     sys_date : integer;
                      VAR tag      ,
                          monat    ,
                          jahr     : integer);

  BEGIN
    tag:=sys_date & $001f;
    monat:=shr(sys_date & $01e0,5);
    jahr:=shr(sys_date & $fe00,9)+1980;
  END;

PROCEDURE inter_time(     sys_time : integer;
                      VAR stunde   ,
                          minute   ,
                          sekunde  : integer);
```

→

```

  BEGIN
    stunde:=shr(sys_time & $f800,11);
    minute:=shr(sys_time & $07e0,5);
    sekunde:=2*(sys_time & $001f);
  END;

PROCEDURE ausgabe_dta(    offset : integer;
                      VAR dta    : dta_type);

  VAR str     : str255;
      cstr    : cstring;
      i       : integer;
      tag     ,
      monat   ,
      jahr    ,
      stunde  ,
      minute  ,
      sekunde : integer;

  BEGIN
    send_string(offset,'');
    WITH dta DO
      BEGIN
        IF reserviert[21]=folder THEN
          send_string(2,'O')
        ELSE
          send_string(2,'');
        i:=0;
        REPEAT
          cstr[i]:=name[i+1];
          i:=succ(i);
        UNTIL (name[i]=chr(0)) OR (i>14);
        cstr[i]:=chr(0);
        ctopstr(cstr,str);
        send_string(14,str);
        writev(str,groesse);
        s_expand(7,str);
        send_string(9,str);
        inter_date(datum,tag,monat,jahr);
        writev(str,tag);
        n_expand(2,str);
        send_string(3,concat(str,'-'));
        writev(str,monat);
        n_expand(2,str);
        send_string(3,concat(str,'-'));
        writev(str,jahr);
        send_string(6,str);
        inter_time(zeit,stunde,minute,sekunde);
        writev(str,stunde);
        n_expand(2,str);
        send_string(3,concat(str,':'));
        writev(str,minute);
        n_expand(2,str);
        send_string(2,str);
        send_signal(lf);
      END;
  END;

PROCEDURE dtopstr(VAR dta : dta_name;
                  VAR str : string);

  VAR i : integer;

  BEGIN
    str:='';
    i:=1;
    WHILE (dta[i]<>chr(0)) AND (i<15) DO
      BEGIN
        str:=concat(str,dta[i]);
        i:=succ(i);
      END;
  END;

PROCEDURE directory(offset ,
                    stype  : integer);

  CONST max_dta = 255;

  TYPE dta_buffer_type = RECORD
                           max : integer;
                           a   : ARRAY
                   [0..max_dta] OF dta_type;
                         END;


  VAR dta_buffer : dta_buffer_type;
      pfad       : string;
      cpfad      : cstring;
      dta_ptr    : dta_ptr_type;
      i          : integer;

  FUNCTION rel(VAR stype : integer;
               VAR op1   ,
                   op2   : dta_type) : boolean;

    VAR str1     ,
        str2     ,
        typ1     ,
        typ2     : string;
        tag1     ,
        tag2     ,
        monat1   ,
        monat2   ,
        jahr1    ,
        jahr2    ,
        stunde1  ,
        stunde2  ,
        minute1  ,
        minute2  ,
        sekunde1 ,
        sekunde2 : integer;

    PROCEDURE gettyp(VAR str ,
                         typ : string);

      VAR ok : boolean;
          i  ,
          j  : integer;

      BEGIN
        ok:=false;
        i:=1;
        REPEAT
          ok:=ok OR (str[i]='.');
          i:=succ(i);
        UNTIL ok OR (i>length(str));
        typ:='';
        IF ok THEN
          FOR j:=i-1 TO length(str) DO
            typ:=concat(typ,str[j]);
      END;

    BEGIN
      IF ((op1.reserviert[21]=folder) AND
         (op2.reserviert[21]=folder)) OR
         ((op1.reserviert[21]<>folder) AND
         (op2.reserviert[21]<>folder)) THEN
        CASE stype OF
          sort_name : BEGIN
                        dtopstr(op1.name,str1);
                        dtopstr(op2.name,str2);
                        rel:=str1<str2;
                      END;
          sort_date : BEGIN
                        inter_date(op1.datum,tag1,
                                   monat1,jahr1);
                        inter_date(op2.datum,tag2,
                                   monat2,jahr2);
                        inter_time(op1.zeit,stunde
                                   1,minute1,sekunde1);
                        inter_time(op2.zeit,stunde
                                   2,minute2,sekunde2);
                        IF jahr1<>jahr2 THEN
                          rel:=jahr1<jahr2
                        ELSE
                          IF monat1<>monat2 THEN
                            rel:=monat1<monat2
                          ELSE
                            IF tag1<>tag2 THEN
                              rel:=tag1<tag2
                            ELSE
                              IF stunde1<>stunde2
                                THEN
                                rel:=stunde1<
                                     stunde2
                              ELSE
                                IF minute1<>
                                   minute2 THEN
                                  rel:=minute1<
```

→

Neue Version

ST-FIBU

Die einfach zu bedienende Finanzbuchhaltung
Professionell – Schnell – Bedienungsfreundlich

- Dialog-orientertes Buchen
- Konten anlegen ganz einfach beim Buchen
- Konten auch mit Namen suchen
- Anzeige vom Monatsjournal am Bildschirm
- Durchsuchen des Monatsjournals
- Berichtigen von Buchungsfehlern im Monatsjournal
- Aktueller Saldo beim Kontoaufruf auf dem Bildschirm
- Buchungsübernahme aus Fakturierung
- Druck aller Listen (Journal, Salden, Kontenplan etc.)
- Bilanz, Gewinn und Verlustrechnung
- Kontenblätter A5 oder A4
- Umsatzsteuervoranmeldung
- Offene-Posten Buchhaltung eingebaut
- Offene-Posten-Liste beim Buchen einsehbar
- Kein Kopierschutz, auch Festplattengeeignet!
- mit Handbuch
- Lauffähig auf jedem ST ab 1 MB und SW-Monitor (SM 124)

DEMO-Version DM 60,00
(wird bei Bestellung angerechnet)

Lern Fibu DM 98,00
für max. 52 Buchungen/Monat

ST-Fibu V. 1.55 DM 398,00/498,00*
für max. 2400 Buchungen/Monat

Zusatzmodul 1 DM 150,00/200,00*
Mahnwesen, Textverarbeitung, Serienbrief

Zusatzmodul 2 DM 200,00/250,00*
Fakturierung

*) mandantenfähig

GMa - Soft
Gerd Matthäus · Betriebswirt
Bergstr. 18 · 6050 Offenbach · Telefon 0 69/89 83 45





Modem Best 2400+ . . 498,–
300, 1200, 1200/75, 75/1200, 1200/1200 (Speeder) 2400 Baud, V21, V22, V22bis, V23, Bell 103 & 212A, Autospeedselection, Hayes kompatibel. *

Modem Best 1–2–3 . . 339,–
300, 1200, 1200/75, 75/1200, 1200/1200 (Speeder) Baud, V21, V22, V23, Bell 103 & 212A, Autospeedselection, Hayes kompatibel. *

Modem Best 1200+ . . 295,–
300, 1200 Baud, V21, V22, Bell 103 & 212A, Autospeedselection, Hayes kompatibel. *

Zusätzlich lieferbar: Modemkarten · Telefaxkarten PC-Mäuse · PC-Handy-Scanner · MNP-Modems.

* Die Inbetriebnahme unserer Modems am öffentlichen Postnetz der BRD einschl. Berlin-West ist verboten und unter Strafe gestellt.
HÄNDLERANFRAGEN ERWÜNSCHT

1 Jahr Garantie

Spitzwegstraße 11 · 4350 Recklinghausen
Tel. (0 23 61) 18 14 85

Prg. für alle ST-Modelle – Exzellent in Struktur, Grafik, Sound – alle Prg. in Deutsch – alle Prg. S/W und Farbe

ATARI ST — TYPIST
Der ST als Schreibmaschine, zeilenweiser Ausdruck, 15zeiliges Bildschirm-Display. Je nach Drucker bis zu 30 Schriften. - File auf Disk, Kopie-Ausgabe DM 86.-

GELD — ATARI ST
30 Routinen für Umgang mit Geld - Anlage Vermögensbildung - Rentensparen -Rendite Zinsen - Kredit - Hypothek - Laufzeit - Amortisation - Raten - Gleitklausel -Effektivzins - Akonto bei Verzinsung -Diskontierung -Konvertierung -Tilgungspläne für alle Modi - Bild-/Druck DM 96.-

ATARI ST — GLOBALER STERNENHIMMEL
Zeigt aktuellen Himmel für jede Zeit/Ort Click auf Obhekt zeigt Namen+Daten -Planeten, Sterne/-bilder - Teleskop - Wandern - Erddrehung - Editor DM 89.-

ASTROL. KOSMOGRAMM — ATARI ST
Auf Namen, Geb.Ort+Zeit werden minutengenau errechnet: Sternzeit, Aszendent MC Zodiakradiant, Position aller Planeten + Sonne, Mond+Mondknoten im Tierkreis,Häuser nach Koch/Schäck, Aspekte - Allgem. Persönlichk.Analyse, Partnerschaftsskala Ausdruck auf 3 DINA4 - Horoskop-Diagramm Koordinatentafel - Kalender DM 75.-

ATARI ST — BIOKURVEN
Wissenschaftl.Trendbestimmung der Körper -Seele-Geist-Rhythmik - Bildschirmausgabe monatlich vor/zurück, aus Drucker beliebig lang, tägl.Analyse + Nennung kritischer Zeiten - Absolut-/Mittelwerte - Wissensch.Grundlagen - Editor f. Zusatzdaten, Grußadresse usw. DM 56.-

KALORIEN-POLIZEI — ATARI ST
Auf Größe, Gewicht, Geschlecht und Arbeitsleistung erfolgen Bedarfsrechnung + Vergleich m.abgefragter Ernährung in Eiweiß-Fett-Kohlenhydraten - Ideal-/Über-/Untergewichtsbestimmung - Vitalstoffe u. Gehalte - Aktivitäten+Verbräuche - Kalorientabelle - Bildschirm-/Druckerausgabe auf einigen DINA4 -Unerbittlich!DM 56.-

****

In Computer Shops oder bei uns per NN + DM 6.- oder Vorkasse + DM 3.-
Preise unverbindlich.
Liste gegen adressierten Freiumschlag DINA5

I. DINKLER



Am Schneiderhaus 7
Tel. 02932/32947 D-5760 ARNSBERG 1

****

ATARI ST — GESCHÄFT
Ein Editor erstellt Formular-, Adressen-Artikel-/Dienstl.Dateien. Die Maus wählt Angebot/Auftrag/A.Bestätigung/Rechnung/-Lieferschein/Mahnung - Eingabe Hand oder Datei - 20 Positionen/DINA4, über Menge, Preis, Aufschlag/Rabatt, Skonto durchgerechnet zur Endsumme - Versand-/Liefer - Verpackungsaussagen -Texteditor DM 196.-

Provisionsabrechnung — ATARI ST
Editor für Vertreter-/Kundenadressen und Formulardaten - Eingabe Hand/Datei - 25 Positionen/DINA4 - Wechselnde Sätze/Pos. Storno, Spesen, Endbetrag/MWSt. DM 116.-

ATARI ST — Inventur, Fibu-gerecht
Kontinuierliche Bestandsverwaltung -Neuerfassung, Streichen, Ändern, Hinzufügen Gruppeninventur nach Code - Jederzeitige Endauswertung m.Druckerausgabe DM 116.-

ETIKETTENDRUCK — ATARI ST
Druckt 40 gängige Haftetiketten-Formate, Auflage nach Wahl, kinderleichte Gestaltung, Ablage für Neuauflage DM 89.-

ATARI ST — Dateiverwaltungen
Datenfelder von 8 Zeilen a 33 Zeichen je Datei max.1000 - Suchcode von max.33 Zch ab 1, mit jedem mehr Zielgr. eingengend - Optionen: Code, Nummer, Blatt vor/zurück Ändern/Streichen/Hinzu - Druck, wo sinnvoll: 80-Zeichen-/Blockliste, Datenmaske Etikett, Zahlenauswertung -Gezielte Aufgabe, schnell am Ziel - Übersichtlich - bedienfreundlich - Keine Blockade !

ADRESSEN	DM 66.-	GALERIE	DM 116.-
BIBLIOTHEK	116.-	LAGER	116.-
BRIEFMARKEN	116.-	PERSONAL	116.-
DISKOTHEK	76.-	STAMMBAUM	116.-
EXPONATE	116.-	VIDEOTHEK	76.-

DEFINDATA, vielseitig verwendbar, zum Selbstdefinieren der Inhalte 146.-

CASINO-Roulett — ATARI ST
Mit Schnellsimulation, Chancentest, Sequenzenverfolgung, Häufigkeitsanalyse - Kassenführung, Setzen durch Anklicken in Bildschirm-Tischgrafik DM 68.-

ATARI ST — BACKGAMMON
Bestechende Grafik, mausgesteuert, ausführliche Anleitung, strategisch DM 58.-

Wenn es um Beratung, Service und gute Preise für Sie geht :

z.B. Software

PHONE DM 79,-
Die bequeme Adressverwaltung um ständig auf dem laufenden zu sein
- als Accessory jederzeit aufrufbar
- schnell durch ausgereifte Programmiertechnik
- Datenaustausch mit anderen Programmen
- minimaler Speicherplatz

ST-DCL DM 298,-
Kommandozeileninterpreter+ Tools
ST-DCL erlaubt es, auf einem Atari unter einer Kommandoumgebung zu arbeiten, die der VAX /VMS-Digital Command Language (DCL) nachempfunden ist. Auch für MS-DOS lieferbar.

PRIVAT-LIQUIDATION

PRAXIS-BUCHHALTUNG

je DM 375,-
Arzt - Software
Aus der Praxis für die Praxis. Praxisliquidation und Praxisbuchhaltung. Info anfordern. Demo plus Handbuch für DM 40,-. (Wird bei Kauf angerechnet).

Sämtliche PD- und Standard-Software lieferbar: Application Systems, C.A.$.H., CCD, DMC, Drews EDV+BTX, G Data, GFA, Hagera, Hyper-Soft, Kniss-Soft, Kieckbusch, NovoPlan, Technobox u.v.m.

z.B. Hardware

vortex HD plus 20	DM	998,-
vortex HD plus 30	DM	1198,-
vortex HD plus 60	DM	1798,-
star LC 10	DM	598,-
NEC P6 plus	DM	1698,-

Wir sind ATARI-Systemhändler.

Alle Drucker mit deutschem Handbuch und Kabel. Sämtliche Hardware für Atari-Computer lieferbar. Alle Angebote freibleibend.

MAUSO ST DM 128,-
Die Mäuse sind los
RS 232 Maus für Atari lieferbar. Die alternative Maus. Inclusive Software.

Hardware Software Service



Computerhandelsgesellschaft mbH
Kölner Str. 132 - 5210 Troisdorf
Telefon (0 22 41) 7 18 97/98

```
                                            minute2
                                    ELSE
                                      rel:=sekunde1<
                                           sekunde2;
                              END;
              sort_size : rel:=op1.groesse>op2.groesse
                          ;
              sort_type : BEGIN
                            dtopstr(op1.name,str1);
                            dtopstr(op2.name,str2);
                            gettyp(str1,typ1);
                            gettyp(str2,typ2);
                            IF typ1<>typ2 THEN
                              rel:=typ1<typ2
                            ELSE
                              rel:=str1<str2;
                          END;
            END
          ELSE
            rel:=op1.reserviert[21]>op2.reserviert[21]
              ;
        END;

  PROCEDURE sort(l ,
                 r : integer);

    VAR i      ,
        j      : integer;
        help   ,
        pivot  : dta_type;

    BEGIN
      WITH dta_buffer DO
        BEGIN
          pivot:=a[(l+r) DIV 2];
          i:=l;
          j:=r;
          REPEAT
            WHILE rel(stype,a[i],pivot) DO
              i:=succ(i);
            WHILE rel(stype,pivot,a[j]) DO
              j:=pred(j);
            IF i<=j THEN
              BEGIN
                help:=a[i];
                a[i]:=a[j];
                a[j]:=help;
                i:=succ(i);
                j:=pred(j);
              END;
          UNTIL i>j;
        END;
      IF l<j THEN
        sort(l,j);
      IF i<r THEN
        sort(i,r);
    END;

  BEGIN
    dta_ptr:=fgetdta;
    IF dgetpath(cpfad,0)=0 THEN
      BEGIN
        ctopstr(cpfad,pfad);
        send_signal(lf);
        send_string(offset,'');
        send_string(50,concat('  Directory : ',
                              pfad));
        send_signal(lf);
        send_signal(lf);
        ptocstr('*.*',cpfad);
        WITH dta_buffer DO
          BEGIN
            max:=-1;
            IF fsfirst(cpfad,normal_file |
                       read_only | folder)>=0 THEN
              REPEAT
                IF dta_ptr^.name[1]<>'.' THEN
                  BEGIN
                    max:=succ(max);
                    a[max]:=dta_ptr^;
                  END;
              UNTIL (max=max_dta) OR (fsnext<>0);
            IF max>-1 THEN
              BEGIN
                sort(0,max);
                FOR i:=0 TO max DO
                  ausgabe_dta(offset,a[i]);
              END
            ELSE
              BEGIN
                send_string(offset,'');
                send_string(50,'  Directory hat
                                  keine Dateien');
                send_signal(lf);
              END;
          END;
      END;
  END;

PROCEDURE x_directory(stype : integer);

 VAR dta_ptr : dta_ptr_type;
     info    : buffer_type;
     cpfad   : cstring;
     str     : str255;

  PROCEDURE dfs(offset : integer);

   VAR cpfad : cstring;
       dstr  : string;
       copy  : dta_type;

   BEGIN
    directory(offset,stype);
     ptocstr('*.*',cpfad);
     IF fsfirst(cpfad,folder)=0 THEN
       WITH dta_ptr^ DO
         REPEAT
           IF (reserviert[21]=folder) AND (name[1
                                        ]<>'.') THEN
             BEGIN
               dtopstr(name,dstr);
               ptocstr(dstr,cpfad);
               IF dsetpath(cpfad)=0 THEN
                 ;
               copy:=dta_ptr^;
               dfs(offset+2);
               dta_ptr^:=copy;
               ptocstr('..',cpfad);
               IF dsetpath(cpfad)=0 THEN
                 ;
             END;
         UNTIL fsnext<>0;
   END;

 BEGIN
   dta_ptr:=fgetdta;
   ptocstr('\',cpfad);
   IF dsetpath(cpfad)=0 THEN
     ;
   send_string(60,concat('Erweiterte Directory
                          des Massenspeichers ',
                         chr(dgetdrive+ord('A')))
                         );
   send_signal(lf);
   send_signal(lf);
   send_string(50,'Label des Speichermediums : ')
              ;
   send_signal(lf);
   send_signal(lf);
   ptocstr('*.*',cpfad);
   IF fsfirst(cpfad,disk_label)=0 THEN
     ausgabe_dta(0,dta_ptr^)
   ELSE
     BEGIN
       send_string(50,'Medium hat keinen Label'
                  );
       send_signal(lf);
     END;
   dfs(0);
   dfree(info,0);
   send_signal(lf);
   writev(str,info[2]*info[3]*info[4]);
   s_expand(7,str);
   send_string(50,concat('Bytes gesamt :',str));
   send_signal(lf);
   writev(str,info[1]*info[3]*info[4]);
   s_expand(7,str);
   send_string(50,concat('Bytes frei   :',str));
   send_signal(lf);
   writev(str,(info[2]-info[1])*info[3]*info[4]);
```

→

```
      s_expand(7,str);
     send_string(50,concat('Bytes belegt :',str));
      send_signal(ff);
    END;

PROCEDURE do_direct;

  VAR button : integer;
      str    : str255;
      ex     : boolean;

  BEGIN
    ex:=false;
    begin_update;
    WITH parameter DO
      REPEAT
        CASE default_sort OF
          sort_name : str:='Name ';
          sort_date : str:='Zeit ';
          sort_size : str:='GröPe';
          sort_type : str:='Typ  ';
        END;
        set_dtext(direct_dialog,tdirect,str,system
                  _font,te_left);
        button:=do_dialog(direct_dialog,0);
        obj_setstate(direct_dialog,button,normal,
                     false);
        CASE button OF
          bname      : default_sort:=sort_name;
          bdatum     : default_sort:=sort_date;
          bgroesse   : default_sort:=sort_size;
          btyp       : default_sort:=sort_type;
          OTHERWISE  : ex:=true;
        END;
      UNTIL ex;
    end_dialog(direct_dialog);
    end_update;
    IF spoolstatus=unused THEN
      CASE button OF
        brekursi   : x_directory(parameter.default_
                                 sort);
        bnormal    : BEGIN
                       directory(0,parameter.
                                 default_sort);
                       send_signal(ff);
                     END;
        OTHERWISE : {dummy};
      END
    ELSE
      dummy:=do_alert('[1][Drucker arbeitet
                       bereits][O.K.]',1);
  END;
```

```
(* resource set indicies for DIRECT *)

CONST
  paramete = 0;    (* form/dialog *)
  paraseit = 3;    (* TEXT in tree PARAMETE *)
  chr1     = 6;    (* TEXT in tree PARAMETE *)
  ord1     = 7;    (* TEXT in tree PARAMETE *)
  makro1   = 8;    (* FTEXT in tree PARAMETE *)
  chr2     = 9;    (* TEXT in tree PARAMETE *)
  ord2     = 10;   (* TEXT in tree PARAMETE *)
  makro2   = 11;   (* FTEXT in tree PARAMETE *)
  chr3     = 12;   (* TEXT in tree PARAMETE *)
  ord3     = 13;   (* TEXT in tree PARAMETE *)
  makro3   = 14;   (* FTEXT in tree PARAMETE *)
  chr4     = 15;   (* TEXT in tree PARAMETE *)
  ord4     = 16;   (* TEXT in tree PARAMETE *)
  makro4   = 17;   (* FTEXT in tree PARAMETE *)
  chr5     = 18;   (* TEXT in tree PARAMETE *)
  ord5     = 19;   (* TEXT in tree PARAMETE *)
  makro5   = 20;   (* FTEXT in tree PARAMETE *)
  chr6     = 21;   (* TEXT in tree PARAMETE *)
  ord6     = 22;   (* TEXT in tree PARAMETE *)
  makro6   = 23;   (* FTEXT in tree PARAMETE *)
  chr7     = 24;   (* TEXT in tree PARAMETE *)
  ord7     = 25;   (* TEXT in tree PARAMETE *)
  makro7   = 26;   (* FTEXT in tree PARAMETE *)
  chr8     = 27;   (* TEXT in tree PARAMETE *)
  ord8     = 28;   (* TEXT in tree PARAMETE *)
  makro8   = 29;   (* FTEXT in tree PARAMETE *)
  chr9     = 30;   (* TEXT in tree PARAMETE *)
  ord9     = 31;   (* TEXT in tree PARAMETE *)
  makro9   = 32;   (* FTEXT in tree PARAMETE *)
  chr10    = 33;   (* TEXT in tree PARAMETE *)
  ord10    = 34;   (* TEXT in tree PARAMETE *)
  makro10  = 35;   (* FTEXT in tree PARAMETE *)
  paraspei = 36;   (* BUTTON in tree PARAMETE *)
  parasync = 37;   (* BUTTON in tree PARAMETE *)
  parazuru = 38;   (* BUTTON in tree PARAMETE *)
  parachec = 39;   (* BUTTON in tree PARAMETE *)
  paravor  = 40;   (* BUTTON in tree PARAMETE *)
  paratest = 41;   (* BUTTON in tree PARAMETE *)
  paraexit = 42;   (* BUTTON in tree PARAMETE *)
  synchro  = 1;    (* form/dialog *)
  synclaen = 3;    (* FTEXT in tree SYNCHRO *)
  syncgrap = 7;    (* FTEXT in tree SYNCHRO *)
  synctext = 8;    (* FTEXT in tree SYNCHRO *
  syncexit = 9;    (* BUTTON in tree SYNCHRO*)
  syncsetz = 10;   (* BUTTON in tree SYNCHR *)
  direct   = 2;    (* form/dialog *)
  bname    = 5;    (* BUTTON in tree DIRECT *)
  bdatum   = 6;    (* BUTTON in tree DIRECT *)
  bnormal  = 7;    (* BUTTON in tree DIRECT *)
  babbdire = 8;    (* BUTTON in tree DIRECT *)
  bgroesse = 9;    (* BUTTON in tree DIRECT *)
  btyp     = 10;   (* BUTTON in tree DIRECT *)
  brekursi = 11;   (* BUTTON in tree DIRECT *)
  tdirect  = 12;   (* TEXT in tree DIRECT *)
```



RR - Soft
Grundstrasse 63
5600 Wuppertal 22
T.: 0202 / 64 03 89

SUPER - STARDRIVER
SUPER NEC-DRIVER

NEUE DRUCKERTREIBER FÜR 1ST WORD PLUS

Nutzen Sie alle Möglichkeiten Ihres Druckers voll aus.
Super-Driver erweitern die Textverarbeitung:
1st Word Plus wird zum DTP-System.

- Zeilenabstände beliebig verändern (z.B. 1 1/2 - zeilig)
- Texte in 2-6 Spalten drucken (echter Mehrspaltendruck)
- alle Druckerschriftarten in einem Dokument einsetzen
- neue Zeichensätze laden und benutzen (z.Zt. 40 Fonts)
- Titel in doppelt und vierfach hohen Zeichen drucken
- gesperrt drucken, Viertelschritt- und Halbschrittaste
- bedienungsfreundliches Handbuch und Referenzkarte

Haben Sie das von 1st Word Plus erwartet?
Warum weniger Leistung, wenn es Superdriver gibt!
Erhältlich für STAR NL10 / LC10 / LC24-10 und NEC P6/P7 / P2200

Test: ST - Computer 4/89 (S.6) und 9/89 (S.11)
Fordern Sie unseren neuen Software-Katalog an:
Große Autoren - Collection, Grafik - Collection,
The Best of Public Domain, Druckerfarbbänder:
STAR LC10 DM 10.-, NEC P6 DM 15.-, P 2200 DM 12.-
Disketten MF 2DD doppelseitig 10 Stck. 22.- DM

35.- DM
+ 5.- DM Versandkosten
Scheine, Scheck oder
Nachnahme (+ 2.-NN-Geb.)

KatCe

Integriertes Entwicklungssystem für alle Atari St Computer.

- Editor
- Pascalcompiler
- Assembler
- Disassembler
- Bibliotheken
- Maschinenmonitor
- Debugger

schneller Compiler (> 400 Zeilen/Sekunde)
viele mathematische Spezialfunktionen
Parallelprozesse
GEM, DOS, BIOS im Sprachumfang
Bitte fordern Sie ein ausführliches Info an!

Preis: 100,- DM
Versandkosten 5,- DM

C. Mayer-Gürr
Software u. Computerbaugruppen
Treptower Str.2
4350 Recklinghausen
Tel. 02361/33153

Pascal

SCANNER

für Atari ST an den Druckern: NEC P2200, NEC P6, NEC P7, EPSON FX 80, FX 85, RX 80, STAR NL10, STAR LC10. (Für weitere Drucker auf Anfrage).

Scannen Sie verwacklungsfrei durch den festen Sitz des Scankopfes. Die Leistungsmerkmale des Scanners:

- Anschluß der Hardware an der RS 232 Schnittstelle. Der empfindlichere Modulport wird nicht belegt. *Kein* Öffnen des Rechners und *keine* Lötarbeiten erforderlich.
- Das **bidirektionale** (!) Scannen bei den Epson Druckern und beim Star LC10 halbiert Ihre Scanzeiten.
- Die Scanroutinen sind in Assemblercode geschrieben und garantieren ein Höchstmaß an Präzision.
- Justierung des Scankontrastes während des Scannens.
- Komfortable Einstellung von Scanparametern.
- Inverses Scannen und Zoomen ist möglich.
- Grafikformate (monochrom): Screen/Doodle-, Degasfomat.

SCANNER (fertig aufgebaut und getestet) mit Software, incl. ausführlicher deutscher Anleitung DM 298,- per NN.

Dipl.-Ing. Gerhard Porada, Dürrlewangstr. 27
7000 Stuttgart 80, ☎ 0711 / 74 47 75.

Schulmeister ST

Atari ST (Mega ST), 500 K Ram sw - Monitor. Die Noten- und Klassenverwaltung mit Pfiff. Ein flexibles, bewährtes Konzept für Lämpels aller Schulstufen. Lassen Sie Ihren Rechner die tägliche Routinearbeit erledigen, damit Sie sich Ihren pädagogischen Aufgaben widmen können. Auch für die Schweiz geeignet!

Ausführliche Information mit Freiumschlag anfordern bei:



M. Heber-Knobloch
Auf der Stelle 27
7032 Sindelfingen

dBMAN

TEL: 089/4 48 06 91
FAX: 089/4 48 38 20

by COMPUTER MAI

Wir belichten Ihre ATARI-Farbgrafiken mit Polaroid Palette.

Wir können folgende Formate verarbeiten:
Art Director, Cyber Paint, Degas, Degas Elite, Imagic und Spectrum 512.

Preise pro Belichtung / 35mm Dia oder Colornegativ:

Anzahl	16 Farben	Spectrum 512
1–5	25 DM	30 DM
6–10	23 DM	27 DM
11–20	20 DM	24 DM
21–∞	16 DM	20 DM

per Nachnahme, zzgl. 5 DM für Porto und Verpackung

Matthias Hans Kurwig
Sassenfeld 71
4054 Nettetal 1
Telefon 02153/89291

Computer Bild

GUTSCHEIN

Nr. St 1 1

Bitte senden Sie mir:

O Kostenlose Info zu PegaFAKT, dem Fakturierungsprogramm mit Lager- und Adressenverwaltung für ATARI ST's, das nur 99,- DM kostet.
(jetzt Version 2.0 mit vielen neuen Funktionen !)

O Die PegaFAKT-DEMO-Diskette für nur 20.- DM (wird beim Programmkauf zurückerstattet)

O Kostenlose Info zu PegaSTIC, dem universellen Etikettendruckprogramm für alle ATARI ST's, das nur 49,- DM kostet.

X Kostenlose Liste mit COMPUTERZUBEHÖR und PROGRAMMEN zu DISCOUNT-PREISEN für:
O C16/64/128 O AMIGA O CPC
O ATARI XL/XE O ATARI ST O PC

Einfach auf Postkarte kleben, den Absender nicht vergessen und ab die Post an:

PegaSoft R. GÄRTIG
SOFTWARE-ENTWICKLUNG & -VERTRIEB
Ringstr.4 7450 Hechingen-Beuren Tel: 07477/8158

Unübersichtlich
wird langsam das PD-Angebot für den Atari ST.

Wenn Sie also keine Lust haben, sich stundenlang durch riesige, nichtssagende Listen zu kämpfen, dann sind Sie bei uns genau richtig!

Unsere PD-Sets enthalten auf je 10 doppelseitigen Qualitätsdisketten jeweils 7 MB, aus 700 Disks ausgewählte, Top-Software. Hier hat der Durchschnitt keine Chance.

PD-Set A (von allem etwas)
PD-Set B (Spiele für S/W)
PD-Set C (Spiele für Farbe)
PD-Set D (Anwendungen)
PD-Set E (Utilities)
PD-Set F (Grafik + Pics)
PD-Set G (Musik, div. Demos)
je **39,-** DM

Sets B – G zusammen (>40 MB) 199,-
(= 35 DM gespart)

Alle Preise incl. Disks, MwSt, Versand- und Verpackungskosten sowie unserer Katalogdiskette. Wir liefern innerhalb von 24h – garantiert.

Gerald Köhler **PD-Schnellversand Atari ST**
Mühlgasse 6 · 6991 Igersheim
Hotline: 07931/43922 + 44661

Scache



mit deutschem Handbuch DM 69,-

Der Disketten und Festplatten Beschleuniger

Diskettenstationen werden bis zu 4000% beschleunigt.

Der Scache läuft mit allen Festplatten und dem neuen TOS 1.4 zusammen. Die Cache-Größe ist von 40KB bis 2MB wählbar. Durch den Einsatz einer sortierten Sektorenliste benötigt der Scache bei 1000 gepufferten Sektoren nur 10 Suchzugriffe, um den gesuchten Sektor zu finden (herkömmliche Cache-Programme benötigen im Mittel 500 Zugriffe).

Scache 4000 DM 79,-

Scache 4000 läuft mit dem neuem AHDI-Treiber, der mit variablen Sektorgrößen arbeitet. Jedes Handbuch wird in 5 Sprachen: Deutsch,Englisch,Französisch,Niederländisch und Spanisch geliefert.

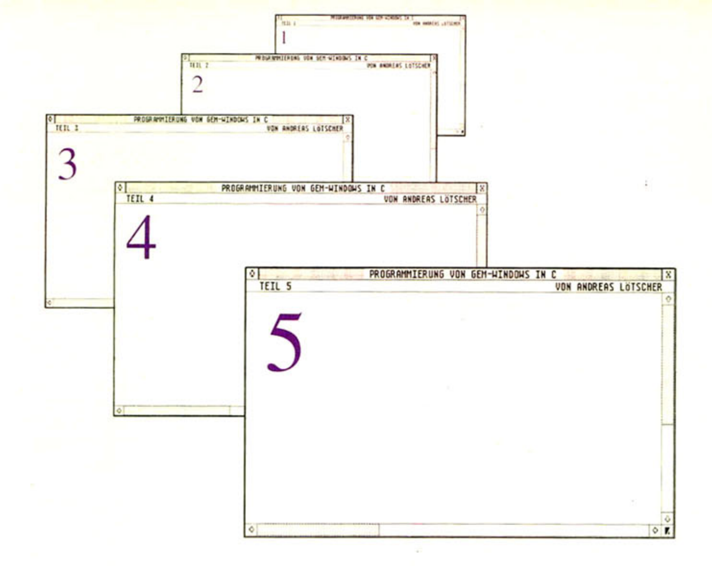
Little Switch

DM 59,-

Der Little Switch teilt Ihren Rechner in bis zu vier "ATARI'S" . Jedem Rechner kann der Speicher in 128KB Schritten zugeteilt werden. Wenn Sie wenig Speicher , dafür aber eine Festplatte besitzen, können Sie einzelne Segmente auf die Festplatte swappen und so mehrere Partitionen anlegen, deren Gesamtgröße größer als die Größe Ihres Speichers ist. Ebenso ist eine Schnittstelle für den Programmierer vorhanden, mit der man einzelne Funktionen des Switchs benutzen kann (Basic, C und Assembler). Jedes Handbuch wird in 5 Sprachen: Deutsch,Englisch,Französisch, Spanisch und Niederländisch geliefert.

Sascha Slavnic ✦ Softwareentwicklung
Wesselburer Weg 15 B
1000 Berlin 27

Tel. 030/436 19 28
431 08 91
✦✦ Nur Versand ✦✦



WINDOWS UNTER GEM

von Rollbalken und Ähnlichem...

Der Titel sagt es bereits, in dieser vorläufig letzten Folge beschäftigen wir uns mit den Bestandteilen eines Fensters, die es uns ermöglichen, mehr Informationen in einem Fenster darzustellen, als eigentlich auf den Bildschirm passen würden. Der "Trick" dabei ist es, nicht alles auf einmal auszugeben, sondern den Benutzer "blättern" zu lassen.

Als erfahrene GEM-Benutzer ist Ihnen dies bestimmt mehr als geläufig: um z.B. in First-Word den Text zu scrollen, greifen Sie zur Maus und klicken einen der Rollpfeile an, verschieben die Rollbox an eine andere Position oder klicken das graue Feld des Rollbalkens an (Vergleichen Sie auch mit Bild 1 aus dem ersten Artikel dieser Folge, dort sehen Sie die Elemente eines Fensters beschrieben). Was geschieht? Wenn Sie ein Rollpfeil anklicken, bewegt sich der Text um eine Zeile nach oben oder unten, bzw. einen Buchstaben nach rechts oder nach links, je nachdem, welchen der vier Rollpfeile Sie angeklickt haben. Wenn Sie den grauen Bereich eines Rollbalkens anklicken, verschiebt sich der Fensterinhalt um eine Seite vorwärts oder rückwärts, bzw. nach links oder rechts, wobei die oberste bzw. unterste Zeile der vorherigen Seite sichtbar bleibt. Ganz anders aber, wenn Sie die Rollbox mit der Maus verschieben. Dann springt der Text an die entprechende Stelle. Es ist so, daß die ganze Länge des Rollbalkens die Länge des Dokumentes, das sich hinter dem Fenster versteckt, darstellt. Die weiße Rollbox steht für den sichtbaren Ausschnitt des Fensters und daher der grau schraffierte Teil für den momentan unsichtbaren Abschnitt des Fensters. D.h., das Fenster stellt eine Art von Sichtfenster dar, durch das Sie einen Ausschnitt des größeren Dokumentes sehen können, je nachdem, über welchem Teil sie es mit den Rollbalken plaziert haben.

Die Realisierung in C

Soweit, so gut. Das Ganze sieht ziemlich einfach aus, jedenfalls für den Benutzer, für den Programmierer stellen sich jedoch einige Schwierigkeiten. Im folgenden werde ich Ihnen also beschreiben, wie ich die Verwaltung der Rollbalken in unsere Windowbibliothek eingebunden habe. Um die zusätzlichen Daten zu verwalten, erfuhr unsere Window-Struktur einige Erweiterungen, außerdem wurde sie in die Headerdatei verlegt. Neu hinzu kamen:

GRECT work; enthält die aktuellen Koordinaten des Arbeitsbereiches (die Fläche innerhalb der Titelzeile (evtl. Infozeile) und den Rollbalken)

WORD scroll_x; Scrollwert für X-Richtung, um diesen Betrag wird der Fensterinhalt bei Betätigung der horizontalen Schieber gerollt. Bei Textausgabe normalerweise 8 (entspricht den Pixeln auf dem Bildschirm)

WORD scroll_y; Scrollwert für Y-Richtung, um diesen Betrag wird der Fensterinhalt bei Betätigung der vertikalen Schieber gerollt. Bei Textausgabe normalerweise 16.

WORD doc_x; Horizontale Position der linken oberen Ecke des Fensters im Dokument (Nicht mit Bildschirmkoordinaten zu verwechseln). Zeigt die linke Spalte z.B. den dritten Buchstaben eines Textes mit X-Scrollweite 8 an, dann beträgt doc_x 16.

WORD doc_y; Vertikale Position der linken oberen Ecke des Fensters im Dokument. Zeigt die oberste Zeile des Fensters z.B. die dritte Zeile eines Textes mit Y-Scrollweite 16 an, dann beträgt *doc_y* 32.

WORD doc_length; Länge des Dokumentes, das das Fenster darzustellen hat. Angabe in Pixeln. Ein Text von 100 Zeilen ist so z.B. bei einer Zeilenhöhe von 16 Pixeln 1600 Pixel lang.

WORD doc_width; Breite des Dokumentes. Ein Text mit der Breite von 65 Buchstaben und 8 Pixeln pro Buchstabe ist daher 520 Pixel breit.

Zu wissen, was für eine Aktion der Benutzer getätigt hat, ist kein Problem. Wenn Sie die Funktion *handle_window()* betrachten, fallen Ihnen drei bisher stiefmütterlich behandelte case-statements auf: *WM_ARROWED*, *WM_HSLID* und *WM_VSLID*. Eine dieser drei Meldungen erhalten wir, wenn wir unser Fenster mit den entsprechenden Elementen beim Öffnen ausgestattet haben und der Benutzer eines davon angeklickt hat. *WM_ARROWED* bedeutet dabei, daß einer der Rollpfeile oder einer der grauen Bereiche angeklickt wurde und der Benutzer entweder eine Zeile oder eine Seite weiter- bzw. zurückblättern, oder nach rechts oder links blättern möchte. *WM_HSLID* meldet GEM bei Verschieben der horizontalen Rollbox, analog *WM_VSLID* bei Verschieben der vertikalen Rollbox.

Um diese Meldungen handzuhaben, schrieb ich entsprechende Funktionen, die die notwendigen Schritte durchführen: Mittels *scroll_wind()* verwalten wir eine *WM_ARROWED*-Message. Der Funktion übergeben wir das handle des Windows *(buffer[3])* und den Inhalt von *buffer[4]*, der die vom Benutzer getätigte Aktion genauer spezifiziert. Vergleichen Sie hierzu das switch-statement in *scroll_wind*. Je nachdem wird dann die Dokumentsposition um den Scrollwert vermindert oder erhöht. Beispiel:

```
case W_UPLINE :
windows[w_handle].doc_y -=
windows[w_handle].scroll_y.
```

Um aber zu verhindern, daß der neue Wert größer als die Dokumentslänge, bzw. kleiner als Null wird, muß eine zusätzliche Überprüfung durchgeführt werden:

```
if(windows[w_handle].doc_y < 0)
windows[w_handle].doc_y = 0
```

bzw.

```
if (windows[w_handlle].doc_y >
windows[w_handle].doc_length)
windows[w_handle].doc_y
windows[w_handle].doc_length
```

Diese Methode funktioniert bei zeilenweisem Rollen, bei seitenweisem Verschieben kommt aber noch etwas hinzu: es wäre sinnvoll, jeweils die letzte Zeile der vorherigen Seite mit anzuzeigen. Eine Seite entspricht hier immer der aktuellen Größe des Arbeitsbereiches. Wir müssen die Dokumentsposition also um die Größe des Arbeitsbereiches plus/minus der X- bzw. Y-Scrollweite vermindern/erhöhen. Beispiel:

```
case W_DNPAGE :
windows[w_handle].doc_y +=
windows[w_handle].work.g_h -
windows[w_handle].scroll_y
```

Die Überprüfung auf Über- oder Unterlauf der maximalen bzw. minimalen Größe bleibt gleich. Nachdem die Dokumentsposition nun so verändert wurde, müssen die Slider (=Rollboxen) neu positioniert werden, da sie ja die Größe und die relative Position des sichtbaren Fensterinhalts zum ganzen Dokument wiedergeben sollen. Hierzu dient *set_slider_pos()*, der wir das Handle des entsprechenden Fensters übergeben. Um die Position der Slider setzen zu können, muß man wissen, daß die GEM-Entwickler die Länge des Rollbalkens auf 1000 festgesetzt haben. Wir müssen also die Dokumentsposition entsprechend umrechnen. Hier die verwendete Formel: Schieberposition = Dokumentsposition * 1000 / (Dokumentsgröße - Arbeitsbereichgröße)

Ein spezieller Fall muß aber berücksichtigt werden: ist die Arbeitsbereichgröße größer als die Dokumentsgröße, dann würde die Schieberposition ungewollterweise negativ, wir müssen diesen Fall also durch eine if-Abfrage abfangen. Nach der Berechnung der neuen Position können wir diesen Wert GEM mittels *wind_set()* mitteilen. Die Implementation sehen Sie in Listing 1.

Für *WM_HSLID* und *WM_VSLID* existieren zwei eigene Funktionen: *wind_hslide()* und *wind_vslide()*. Beide berechnen im Prinzip dasselbe, nur entweder horizontal oder vertikal, nämlich die neue Dokumentsposition. Dies ist der umgekehrte Fall von vorhin, bei *set_slider_pos()* war die Dokumentsposition bekannt, und die neue Sliderposition mußte berechnet werden. Jetzt ist die Sliderposition *(buffer[4])* bekannt, und wir müssen die neue Dokumentspositon ermitteln. Aus der oben genannten Formel läßt sich dies leicht ableiten: Dokumentsposition = Schieberpositon * (Dokumentsgröße - Arbeitsbereichsgröße)/1000. Schließlich darf man nicht vergessen, die neue Sliderposition mittels *wind_set()* zu setzten, da dies GEM nicht selbstständig macht. Zu allerletzt, jetzt wieder in *handle_window()*, wird das ganze Fenster neu gezeichnet, da sich ja der Inhalt verschoben hat. Dazu rufen wir *full_redraw()* auf, das nichts anderes tut, als den Fensterinhalt mittels *clear_window()* zu löschen und je nach Fenster die entsprechende Redraw-Routine aufzurufen.

Sehr schön, aber...

... das ist noch nicht alles. Einerseits müssen wir *open_window()* anpassen, andererseits darf man keinesfalls vergessen, daß, wenn das Fenster in der Größe verändert wird, sich die Schiebergrößen ändern, da die Rollbalkenlänge verändert wird. Zunächst aber zu der nochmals erweiterten *open_window()*-Funktion. Notwendigerweise müssen wir ihr die X- und Y-Scrollwerte und die Dokumentslänge und Dokumentsbreite mitteilen. Sie trägt dann diese Werte in die Struktur ein. Außerdem setzt sie die horizontale und vertikale Dokumentsposition auf Null und berechnet die aktuellen Arbeitsbereichkoordinaten durch einen direkten Aufruf von *wind_calc()*. Jetzt zur Änderung der Schieber bei einer Größenänderung des Fensters: am Ende der Anweisungen, die in *handle_window()* bei *WM_MOVED* aufgerufen werden (d.h. wenn das Fenster in der Größe verändert wurde) fügen wir einen Aufruf von *wind_calc_work()* und *set_slider_size()* ein. *Wind_calc_work()* berechnet aus den Koordinaten des gesamten Fensters diejenigen des Arbeitsbereiches (wie dies schon *open_window()* tat) und trägt die neuen Daten in die Struktur ein. *Set_slider_size()* berechnet, wie der Name schon sagt, die neue Größe der Slider und teilt diese Werte GEM mit, das dann die Slider neu zeichnet. Desweiteren müssen wir auch daran denken, daß bei einer Betätigung des Volle-Größen-Ecks die Fenstergröße verändert wird. Daher ergänzen wir *handle_full()* durch einen *wind_calc_work()* Aufruf und fügen in *handle_window()* nach dem Aufruf von *handle_full()* einen Aufruf von *set_slider_size()* ein.

Endlich

Damit möchte ich diese Folge vorläufig beenden. Sie haben nun eine praktische Windowbibliothek in den Händen, die es Ihnen ermöglicht, Fenster in GEM einfach handzuhaben. Natürlich steht es Ihnen frei, diese Ihren Wünschen anzupassen und zu erweitern - Möglichkeiten gäbe es noch genug...

Andreas Lötscher

```
/* HEADERDATEI, DIE WICHTIGE KONSTANTEN FÜR DIE
   WINDOW-ROUTINEN ENTHÄLT
   (vgl. Gem-Bibliothek ihres C-Compilers !)    */



/* Definitionen für wind_create()   */

#define NAME    0x0001
#define CLOSER  0x0002
#define FULLER  0x0004
#define MOVER   0x0008
#define INFO    0x0010
#define SIZER   0x0020
#define UPARROW 0x0040
#define DNARROW 0x0080
#define VSLIDE  0x0100
#define LFARROW 0x0200
#define RTARROW 0x0400
#define HSLIDE  0x0800

/* Definitionen für wind_get() bzw. wind_set() */

#define WF_KIND         1
#define WF_NAME         2
#define WF_INFO         3
#define WF_WORKXYWH     4
#define WF_CURRXYWH     5
#define WF_PREVXYWH     6
#define WF_FULLXYWH     7
#define WF_HSLIDE       8
#define WF_VSLIDE       9
#define WF_TOP          10
#define WF_FIRSTXYWH 11
#define WF_NEXTXYWH  12
#define WF_RESVD        13
#define WF_NEWDESK      14
#define WF_HSLSIZE      15
#define WF_VSLSIZE      16
#define WF_SCREEN       17

/* Definitionen für wind_update()  */

#define END_UPDATE 0
#define BEG_UPDATE 1
#define END_MCTRL  2
#define BEG_MCTRL  3

/* Definitionen fur wind_scroll()  */

#define W_UPPAGE   0
#define W_DNPAGE   1
#define W_UPLINE   2
#define W_DNLINE   3
#define W_LFPAGE   4
#define W_RTPAGE   5
#define W_LFLINE   6
#define W_RTLINE   7

/* Definitionen für wind_calc()     */

#define WC_BORDER 0
#define WC_WORK   1

/* Definitionen für evnt_mesag()    */

#define MU_KEYBD   0x0001
#define MU_BUTTON 0x0002
#define MU_M1      0x0004
#define MU_M2      0x0008
#define MU_MESAG   0x0010
#define MU_TIMER   0x0020

/* von evnt_mesag() gemeldete Aktionen */

#define MN_SELECTED 10
#define WM_REDRAW   20
#define WM_TOPPED   21
#define WM_CLOSED   22
#define WM_FULLED   23
#define WM_ARROWED  24
#define WM_HSLID    25
#define WM_VSLID    26
#define WM_SIZED    27
#define WM_MOVED    28
#define WM_NEWTOP   29
#define AC_OPEN     40
#define AC_CLOSE    41

/* Definitionen für graf_growbox()
   bzw. für graf_shrinkbox()          */

#define FMD_START  0
#define FMD_GROW   1
#define FMD_SHRINK 2
#define FMD_FINISH 3

/* Strukturdefinition für Rechteckslisten  */

typedef struct grect
{
  short g_x;
  short g_y;
  short g_w;
  short g_h;
} GRECT;

typedef
struct wind_data /*Struktur, die Wissenswertes*/
{               /*ueber unsere Fensterenthaelt */
  char  name[80]; /* Fenstername               */
  GRECT max;       /* Maximalgroesse            */
  GRECT work;      /* Arbeitsbereichgroesse     */
  WORD  elements; /* Bestandteile des Fensters*/
  WORD  align; /* Faktor zur hor. Ausrichtung */
  WORD  snap;  /* Fenster snappen (TRUE/FALSE)*/
  WORD  full;       /* Full-Flag (TRUE/FALSE)    */
  WORD  scroll_x; /* Scrollwert fuer X-Achse  */
  WORD  scroll_y; /* Scrollwert fuer Y-Achse  */
  WORD  doc_x;    /* X-Position des Dokumentes */
  WORD  doc_y;    /* Y-Position des Dokumentes */
  long  doc_length;     /* Dokumentslaenge      */
  WORD  doc_width;      /* Dokumentsbreite      */
  void  (*w_redraw)(); /* Pointer auf Redraw-
                               Funk.             */
} WIND_DATA;

/* Ein Makro, das einen 32 Bit Pointer-Wert als
   zwei
   16 Bit Werte zurückgibt.
   Wird in wind_set() gebraucht, um die Adresse
   des
   Fensternamens zu übergeben                 */

#define ADDR(a) ((long)(a) >> 16), ((long)(a)
                 & 0xFFFF)

/* Makros, die zwei Zahlen vergleichen und den
   groesseren bzw. kleineren zurueckgeben */

#define _max(a,b) ((a)>(b)?(a):(b))
#define _min(a,b) ((a)<=(b)?(a):(b))
```

```
       /* Window-Bibliothek-Erweiterung          */
      /* Andreas Loetscher 1988                  */
     /* verwendeter Compiler : Lattice C 3.04 */
    /*****************************************/

#include <a:\headers\portab.h>
#include <c:\listing1.h>

WORD    ap_id, handle, work_in[12], work_out[57];
WIND_DATA windows[8];

void gem_init()
{
}

void gem_exit()
{
}

void
clear_window(w_hndl)
{
}

rc_intersect(p1,p2)
{
```

→

```
}


void          /* Setzen der Groesse der Slider */
set_slider_size(w_handle)
  WORD w_handle;
{ long h_size, v_size;

  h_size = windows[w_handle].work.g_w * 1000 /
           windows[w_handle].doc_width;
  v_size = windows[w_handle].work.g_h * 1000 /
           windows[w_handle].doc_length;
  wind_set(w_handle,WF_HSLSIZE,h_size,0,0,0);
  wind_set(w_handle,WF_VSLSIZE,v_size,0,0,0);
}


void         /* Setzen der Position der Slider */
set_slider_pos(w_handle)
  WORD  w_handle;
{ long  x_pos, y_pos;

  if(windows[w_handle].doc_width <=
     windows[w_handle].work.g_w)
       x_pos = 0;
  else x_pos = windows[w_handle].doc_x * 1000 /
               ( windows[w_handle].doc_width -
                 windows[w_handle].work.g_w );
  if(windows[w_handle].doc_length <=
     windows[w_handle].work.g_h)
       y_pos = 0;
  else y_pos = windows[w_handle].doc_y * 1000 /
               ( windows[w_handle].doc_length -
                 windows[w_handle].work.g_h );
  wind_set(w_handle,WF_HSLIDE,x_pos,0,0,0);
  wind_set(w_handle,WF_VSLIDE,y_pos,0,0,0);
}


/* erweiterte open_window()-Funktion ...      */

WORD open_window(w_name,redraw,was,algn,snp,
                 s_x,s_y,doc_l,doc_w,
                 x1,y1,w1,h1,
                 mx,my,mw,mh)
  char    *w_name;      /* Ptr auf Namensstring*/
  void    (*redraw)(); /* Ptr auf Redraw-
                          Funktion            */
  WORD    was,          /* Liste der Elemente  */
          algn,snp,     /* align-Wert, snap j/n*/
          s_x,s_y,      /* Scrollwerte X/Y     */
          doc_l,doc_w,  /* Dokumentslaenge/
                           breite              */
          x1,y1,w1,h1,  /* Startkoordinaten    */
          mx,my,mw,mh;  /* Maximalkoordinaten  */
                        /* falls mw==0 gleich  */
                        /* Desktop             */
{
  WORD    w_handle;

  if(mw==0)
    /* ermittelt die Groesse des Desktop:      */
    wind_get(0,4,&mx,&my,&mw,&mh);

    /* und meldet ein Fenster an               */
  w_handle=wind_create(was,mx,my,mw,mh);

    /* traegt wichtige Daten in Struktur ein :*/
  windows[w_handle].max.g_x  = mx;
  windows[w_handle].max.g_y  = my;
  windows[w_handle].max.g_w  = mw;
  windows[w_handle].max.g_h  = mh;
  windows[w_handle].elements = was;
  windows[w_handle].align    = algn;
  windows[w_handle].snap     = snp;
  windows[w_handle].w_redraw = redraw;
  windows[w_handle].scroll_x = s_x;
  windows[w_handle].scroll_y = s_y;
  windows[w_handle].doc_length = doc_l;
  windows[w_handle].doc_width  = doc_w;
  windows[w_handle].doc_x =
  windows[w_handle].doc_y = 0;
  strcpy(windows[w_handle].name,w_name);
  windows[w_handle].full =
      (x1==mx && y1==my && w1==mw && h1==mh);

  wind_calc(WC_WORK,was,x1,y1,w1,h1,
                  &windows[w_handle].work.g_x,
                  &windows[w_handle].work.g_y,
                  &windows[w_handle].work.g_w,
                  &windows[w_handle].work.g_h);

    /* setzen des Fensternamens :            */
  wind_set(w_handle,2,
           ADDR(windows[w_handle].name),
           0,0);
    /* zeichnet öffnende Box:                */
  graf_growbox(0,0,0,0,x1,y1,w1,h1);

  wind_open(w_handle,x1,y1,w1,h1);
  clear_window(w_handle);
  set_slider_size(w_handle);
  set_slider_pos(w_handle);

    /* schliesslich geben wir dem rufenden
       Programm die Identifikationsnummer
       des Fensters zurück:              */
  return(w_handle);
}


void    /* Berechnung der Arbeitsflaeche eines */
wind_calc_work(w_handle,x,y,w,h) /* Fensters   */
  WORD w_handle,x,y,w,h;
{
  wind_calc(WC_WORK,windows[w_handle].elements,
         x,y,w,h,&windows[w_handle].work.g_x,
                  &windows[w_handle].work.g_y,
                  &windows[w_handle].work.g_w,
                  &windows[w_handle].work.g_h);
}


WORD
rc_equal(p1,p2)
{
}

void
handle_full(w_hndl)
{
  ...
  if (rc_equal(&curr, &full))
  { ...
   wind_set(w_hndl,WF_CURRXYWH,prev.g_x,prev.g_y,
                               prev.g_w,prev.g_h);
    windows[w_hndl].full = FALSE;
    wind_calc_work(w_hndl,prev.g_x,prev.g_y,
                          prev.g_w,prev.g_h);
  }
  else
  { ...
   wind_set(w_hndl,WF_CURRXYWH,full.g_x,full.g_y,
                               full.g_w,full.g_h);
    windows[w_hndl].full = TRUE;
    wind_calc_work(w_hndl,full.g_x,full.g_y,
                          full.g_w,full.g_h);
  }
}

void
snap(w1,w2)
{
}

WORD
align(k,n)
{
}

void    /* zeichnet ganzes Fenster neu         */
full_redraw(w_handle)
  WORD  w_handle;
{
  clear_window(w_handle);
  (*windows[w_handle].w_redraw)();
}

do_redraw(buffer)
{
}

```

→

```
void        /* Berechnen der Dokumentsposition */
wind_hslide(w_handle,newpos) /* -> horizontal */
  WORD w_handle,newpos;
{
  windows[w_handle].doc_x = newpos *
           (windows[w_handle].doc_width -
            windows[w_handle].work.g_w) / 1000;
  wind_set(w_handle,WF_HSLIDE,newpos,0,0,0);
}


void
wind_vslide(w_handle,newpos) /* -> vertikal    */
  WORD w_handle,newpos;
{
  windows[w_handle].doc_y = newpos *
           (windows[w_handle].doc_length -
            windows[w_handle].work.g_h) / 1000;
  wind_set(w_handle,WF_VSLIDE,newpos,0,0,0);
}


void                /* Slidermanager             */
scroll_wind(w_handle,what)
  WORD  w_handle, what;
{
  switch(what)
  {
    case W_UPPAGE : windows[w_handle].doc_y -=
                    windows[w_handle].work.g_h +
                    windows[w_handle].scroll_y;
                    if(windows[w_handle].doc_y<0)
                      windows[w_handle].doc_y =0;
                    break;
    case W_DNPAGE : windows[w_handle].doc_y +=
                    windows[w_handle].work.g_h -
                    windows[w_handle].scroll_y;
                    if(windows[w_handle].doc_y >
                    windows[w_handle].doc_length)
                    windows[w_handle].doc_y =
                    windows[w_handle].doc_length;
                    break;
    case W_UPLINE : windows[w_handle].doc_y -=
                    windows[w_handle].scroll_y;
                    if(windows[w_handle].doc_y<0)
                    windows[w_handle].doc_y = 0;
                    break;
    case W_DNLINE : windows[w_handle].doc_y +=
                    windows[w_handle].scroll_y;
                    if(windows[w_handle].doc_y >
                    windows[w_handle].doc_length)
                    windows[w_handle].doc_y =
                    windows[w_handle].doc_length;
                    break;
    case W_LFPAGE : windows[w_handle].doc_x -=
                    windows[w_handle].work.g_w +
                    windows[w_handle].scroll_x;
                    if(windows[w_handle].doc_x<0)
                    windows[w_handle].doc_x = 0;
                    break;
    case W_RTPAGE : windows[w_handle].doc_x +=
                    windows[w_handle].work.g_w -
                    windows[w_handle].scroll_x;
                    if(windows[w_handle].doc_x >
                    windows[w_handle].doc_width)
                    windows[w_handle].doc_x =
                    windows[w_handle].doc_width;
                    break;
    case W_LFLINE : windows[w_handle].doc_x -=
                    windows[w_handle].scroll_x;
                    if(windows[w_handle].doc_x<0)
                      windows[w_handle].doc_x =0;
                    break;
    case W_RTLINE : windows[w_handle].doc_x +=
                    windows[w_handle].scroll_x;
                    if(windows[w_handle].doc_x >
                    windows[w_handle].doc_width)
                    windows[w_handle].doc_x =
                    windows[w_handle].doc_width;
                    break;
  }
  set_slider_pos(w_handle);
}


void                  /* allg. Fensterverwaltung    */
handle_window(buffer)
{

  switch(buffer[0])
  {
    ...
    case WM_FULLED  : handle_full(buffer[3]);
                      set_slider_size(buffer[3]);
                      break;
    case WM_ARROWED : scroll_wind(buffer[3],
                                  buffer[4]);
                      full_redraw(buffer[3]);
                      break;
    case WM_HSLID   : wind_hslide(buffer[3],
                                  buffer[4]);
                      full_redraw(buffer[3]);
                                  break;
    case WM_VSLID   : wind_vslide(buffer[3],
                                  buffer[4]);
                      full_redraw(buffer[3]);
                                  break;
    case WM_SIZED   :
    case WM_MOVED   :
                    ...
                  wind_calc_work(buffer[3],
                                 buffer[4],
                       buffer[5],buffer[6],
                                 buffer[7]);
                  set_slider_size(buffer[3]);
                                 break;
    ...
  }
}

```

```
       /* Demonstrationsprogramm zum Snappen  */
      /* von Windows                          */
     /* Andreas Loetscher 1988                */
    /* Compiler : Lattice-C 3.04             */
   /***************************************/

#include <a:\headers\portab.h>
#include <c:\listing1.h>

extern WORD handle;
extern WIND_DATA windows[];
WORD   w_handle;

char   *text[] = {"Dies soll ein Beispieltext
                   sein",
                  "der die Funktion der Slder,",
                  "und der Rollpfeile
                   demonstrieren",
                  "soll.",
                  "",
                  "Sie koennen ihn beliebig
                   umherschieben...",
                  "",
                  "Die Geschwindigkeit der
                   Textausgabe ist",
                  "nicht gerade grossartig, aber
                   das ganze",
                  "ist auch noch nicht
                   optimiert..."
                 };

void
output()
{
  WORD    x,y,i;

  x = windows[w_handle].doc_x +
      windows[w_handle].work.g_x;
  y = windows[w_handle].doc_y +
      windows[w_handle].work.g_y+16;
  for(i=0; i<10; i++,y += 16)
      v_justified(handle,x,y,text[i],200,0,0);

}

void
main()
```

→

```
{
  WORD  buffer[8];

  gem_init();
  graf_mouse(0,0);

  w_handle= open_window("Demofenster",
                        output,
                        NAME+CLOSER+MOVER+
                        SIZER+FULLER+
                        UPARROW+DNARROW+
                        VSLIDE+HSLIDE+
                        LFARROW+RTARROW,
                        8,TRUE,
                        8,16,
                        800,640,
                        40,30,220,290,
                        16,35,608,349);
  do
  {
    evnt_mesag(buffer);
    handle_window(buffer);
  }while(buffer[0] != WM_CLOSED);

  wind_delete(w_handle);
  gem_exit();
}


```



Künstliche neurale Netze

Seit Urbeginn gehört zum menschlichen Denken das Nachdenken über das Denken und seine Rückverbundenheit zum Organismus. Und doch ist das Korrelat zwischen Denken und Gehirn erst relativ spät entdeckt worden: "Ist es das Blut, mit dem wir denken, oder die Luft oder das Feuer?", fragte Alkmaion von Kroton (etwa 500 v. u. Z.) und antwortete: "Solange das Gehirn unversehrt ist, solange hat auch der Mensch seinen Verstand ... Daher behaupte ich, daß es das Gehirn ist, das den Verstand sprechen läßt." (1)

Zweieinhalbtausend Jahre später scheint die alleinige Herrschaft des Gehirns gebrochen, künstliche Systeme sollen ihm gleichgestellt werden: "Für die Behauptung, daß Computer nicht intelligenter sein können als Menschen, gibt es kein ernsthaftes Argument ... In zehn, zwanzig Jahren wird es darüber keine ernsthafte Diskussion mehr geben", diagnostizierte Prof. Karl Steinbuch zu Beginn der siebziger Jahre (2). Nun, die Zeit ist gekommen, immer schnellere und leistungsfähigere Computer sind entstanden, trotzdem oder gerade deshalb hat das klare Bild von der prinzipiellen Vergleichbarkeit zwischen Gehirn und Computer viel von seiner vermeintlichen Brillanz eingebüßt, wird zunehmend unschärfer.

'Die KI ist tot; es lebe die KI!' Ganz nüchtern betrachtet, die klassische Wissenschaftsrichtung der 'Künstlichen Intelligenz' ist - trotz mancher Erfolge - an ihren Ansprüchen gemessen, in einer Sackgasse gelandet. Den Weg in die Zukunft soll nunmehr ein neuer Forschungszweig antreten. Er firmiert unter dem Oberbegriff 'Konnektionismus', eine Bezeichnung, die anerkennt, daß Gehirne nicht wie Zentralprozessoren funktionieren, sondern ihre Kapazität aus dem Verbundverhalten von Nervenzellen ziehen. Infolgedessen, ausgehend von anderen Voraussetzungen und in Anspruchnahme anderer Mittel, insbesondere der sogenannten 'neuralen Netze' (englisch: 'neural nets', im Deutschen meist als 'neuronal' bezeichnet), hoffen die Abtrünnigen, die alten hochgesteckten Ziele zu erreichen.

Wie bei der konventionellen KI strebt man wiederum nichts Geringeres an, als den Aufbau und die Leistungen des menschlichen Gehirns künstlich nachzubilden. Früher hieß es, das Gehirn funktioniere wie ein sequentiell arbeitender Computer, heute berufen sich die Forscher auf Parallelrechner. Früher versuchte man die Fähigkeiten des Gehirns mit Hilfe heuristischer Verfahren nachzuahmen, heute gelten die selbstlernenden neuralen Netze als Wundermittel. Eine Wandlung, die nicht von ungefähr kommt. Schon zeichnen sich bei den hochgepriesenen Expertensystemen der klassischen KI deutlich die Grenzen ihrer Leistungsfähigkeit ab. So kommen unabhängig voneinander Gutachter vom Institut für Medizinische Informatik und Systemforschung (MEDIS), vom Fraunhofer-Institut und vom Institut für Sozialwissenschaftliche Forschung für ihren jeweiligen Bereich zu dem Ergebnis, daß eine breite Nutzung der Expertensysteme zukünftig eher unwahrscheinlich sein wird (3).

Das Perceptron

Dabei waren es die Expertensysteme, von denen die frühen Ansätze zu neuralen Netzen abgelöst wurden. Bereits 1958 führte der Sozialwissenschaftler Frank Rosenblatt im Cornell Aeronautical Laboratory die Computersimulation eines neuralen Netzes vor: 'Perceptron' benannt - als Wahrnehmungs- (englisch perception) und darüberhinaus als Erkenntnismaschine gedacht. Auf einem der damals schnellsten Rechner, einem IBM 704, brauchte das Simulationsprogramm eine halbe Stunde für die Abtastung eines Viereckmusters, die das fertiggestellte Perceptron in Bruchteilen von Sekunden hatte ausführen sollen.

Zwei Jahre später konnte man den ersten Prototypen des Perceptrons bewundern. In diesem Apparat war jede der 400 Photozellen, die in ihrer Gesamtheit die Retina des Auges versinnbildlichten, zuständig für einen Ausschnitt eines abzutastenden Bildes mit Buchstaben. Jeweils mehrere Photozellen leiteten die aufgenommenen Informationen als 0- oder 1-Signale zu 512 Aussageeinheiten (predicates), welche verschiedene lineare Funktionen berechneten. Von hier führten Verbindungen zu Entscheidungseinheiten, die

abschließend eine Ja- oder Nein-Aussage über die abgetasteten Muster lieferten.

Bei richtigen Entscheidungen der Maschine griff man nicht ein, ansonsten veränderte ein menschlicher Operator die elektrischen Schaltungen. Halbautomatisch nach dem 'Versuch-und-Irrtum-Prinzip' arbeitend, war das Perceptron zu einfachsten Klassifikationsaufgaben fähig. Nach Abschluß der Lernphase gelang es dem Apparat, einige neue Buchstaben zu identifizieren, aber nur dann, wenn sie vollständig und in der gleichen Größe wie die vorher gelernten dargeboten wurden.

Ein für damalige Verhältnisse erstaunliches Ergebnis. Und es sollte ausreichen, die grundsätzliche Übereinstimmung zwischen Gehirn und Maschine zu postulieren. Schon glaubte Rosenblatt, mit seiner Maschine irgendwann die Werke Shakespeares interpretieren zu können, und der "New Yorker", das Perceptron im Visier, verstieg sich zu der Formulierung: "Unser eigenes Gehirn vermag die Dinge nicht zu fassen, die die Maschine leisten soll." (4)

Die Zeit verfloß: Obwohl hunderte von Forschern mit ähnlichen Apparaturen arbeiteten, wurde die Kluft zwischen Anspruch und Wirklichkeit immer größer, und 1969 schließlich versetzten Marvin Minsky - der 1954 in seiner Dissertation selbst ein neurales Netz aus 400 Vakuumröhren vorgestellt hatte - und Seymour Papert dem Perceptron den Todesstoß. In einem eigens zu diesem Thema geschriebenen Buch, "Perceptrons", deckten sie Punkt für Punkt die Schwächen der Rosenblattschen Maschine auf. Beispielsweise konnten sie zeigen, daß ein solches System, wenn in ihm die Eingangsneuronen direkt mit Ausgangsneuronen verbunden sind, nicht einmal in der Lage ist, die Anzahl der Eingangs-Einsen auf gerade oder ungerade zu testen (XOR-Problem, s.u. und Listing), wie sollte es dann komplexe Aufgaben bearbeiten? Eine Lösung schien nicht in Sicht, denn einfach die Zahl der Neuronen zu erhöhen, um die Unzulänglichkeit aufzuheben, dies würde - laut Minsky und Papert - in einem unübersehbaren Gewirr enden, in dem nicht eindeutig bestimmt werden könnte, welche der Verbindungen des Netzes bei einem Fehlverhalten nachzustellen sei.

In der Folge verlor die Perceptronforschung an Bedeutung; es begann das Zeitalter der Expertensysteme. 1982 setzte eine neue Phase in der Beschäftigung mit neuralen Netzen ein. Anstoß dazu gab der Festkörperphysiker John Hopfield vom California Institute of Technology (Caltech).

Modelle und Wirklichkeit

Alle unsere Vorstellungen über die Welt besitzen Modellcharakter. Wahrscheinlich werden wir nie wissen, wie die Welt 'wirklich' ist, wir erleben und erkennen die Welt nur so, wie sie unsere Sinnesorgane und unser Verstand mit Hilfe seiner Werkzeuge (darunter die Mathematik) erfassen. Wir rekonstruieren mithin die Welt, unbewußt in der Wahrnehmung, bewußt und kritisch in der Wissenschaft (5). Dabei wird permanent abstrahiert und generalisiert, um grundlegende Strukturen und Funktionen hypothetisch herauszuarbeiten. Selbstverständlich müssen sich diese Hypothesen auf ihre Angemessenheit und Brauchbarkeit prüfen und gegebenenfalls korrigieren lassen.

Abb. 1: Energetische Landschaft nach Tank und Hopfield (8)

Kein Geringerer als John von Neumann war sich im klaren darüber, daß die Sprache des Gehirns mit einer anderen als der uns zur Zeit verfügbaren Mathematik beschrieben werden müßte. Im gleichen Sinne argumentiert der Verhaltensforscher John C. Lilly, wenn er konstatiert: "Hätten wir am Anfang (statt unserer Arithmetik) etwas entdeckt, was patenter, intellektueller, der Mathematik des Gehirns analoger sei, dann wären wir viel weiter gekommen." (6) Wie aber sollte man auskommen, ohne die verwirrenden, unzureichenden, oft widersprüchlichen Ergebnisse aus der Neurologie zu verwerten?

Ganz einfach, indem aus der mathematischen Physik Erklärungsmuster übernommen wurden. Hopfield zum Beispiel zog Parallelen zur Funktionsweise des Gehirns aus den Speichereigenschaften von 'Spingläsern'. Dies sind Legierungen, in denen ungeordnete magnetische Kräfte unter bestimmten Bedingungen dem Lernverhalten ähnliches zeigen. Gerade thermodynamische Konzepte (darunter fällt Hopfields Neuronen-Modell) spielen eine nicht zu übersehende Rolle bei der Erklärung neuraler Netzwerke.

Für die traditionellen Vertreter der KI repräsentierte das Gehirn nur eine von vielen Verwirklichungen denkender Maschinen, nicht einmal eine besonders leistungsfähige. Erst die Methoden der KI, glaubten sie, würden zu logischen Maschinen führen, die im Gegensatz zum natürlichen Denkapparat alle berechenbaren Funktionen auch erfolgreich berechnen könnten. Ein Irrglaube, wie sich herausstellte. Immerhin hat die KI dazu beigetragen, bewußt zu machen, mit welcher Leichtigkeit das Gehirn Probleme löst, die vordem gar nicht als solche wahrgenommen wurden. Notgedrungen richtet sich der Blick wieder zurück auf das eigene Erkenntnisorgan, versucht man dessen Methoden zu ergründen und zu simulieren: erneut mit künstlichen neuralen Netzen. Unmißverständlich gibt Valentin Braitenberg, Direktor am Max-Planck-Institut für biologische Kybernetik, die Richtung vor: "Verstehen heißt heute, im Computerzeitalter auch: bauen können." (7)

Bislang kann kein Wissenschaftler mit verläßlicher Sicherheit den Zusammenhang zwischen neuraler Informationsverarbeitung und Denken angeben, gleichwohl aber will man neue Ideen verwirkli-

chen. So greift man auf Modelle aus der Physik und auf grobe Hypothesen aus der Neurologie der 40er Jahre zurück: auf das Neuronenmodell von Warren S. McCulloch und Walter H. Pitts, das sich aus einfachen Schwellenwertelementen zusammensetzt und auf die theoretische Lernregel von Donald O. Hebb, die eine Verstärkung der Verbindung zwischen zwei Neuronen vorsieht, wenn beide zur gleichen Zeit aktiv sind. Mit diesen Hypothesen läßt sich arbeiten, in der Gewißheit allerdings, am natürlichen Nervengeflecht vorbeizuforschen.

Gegenwärtig finden wir daher nur wenige Vertreter des Konnektionismus, die behaupten, mit ihren neuralen Netzwerk-Modellen realistische Kopien der neuralen Verflechtungen des Gehirns erstellt zu haben, dennoch soll bereits die Terminologie eine Äquivalenz zwischen Maschi-

Informationsverarbeitung im Gehirn

Beim Vergleich zwischen Gehirn und Computer werden drei Tatsachen gern übergangen, daher sollen sie hier vorweg aufgeführt werden: Zum einen, das Gehirn ist keine tote Maschine, es lebt; zum zweiten, das Gehirn ist kein elektrisches, sondern ein chemisches Organ; und zum dritten, es gibt kein emotionsfreies Denken.

Beschränken wir uns im folgenden auf die Beschreibung der Neuronen (9). Sie stellen im Nervensystem die kleinste morphologische und funktionelle Einheit dar. Eingeteilt nach Größe, Auftreten im Gehirn, Verzweigungsart lassen sich bis zu 1000 Kategorien finden, in welche Neuronen eingruppiert werden können. Wesentlich schneller und einfacher unterscheidet man jedoch Neuronen danach, ob sie mit Sinneszellen gekoppelt sind, also sensorische Funktionen übernehmen, oder mit Muskelzellen in Verbindung stehen, also motorische Funktionen ausüben. Neuronen, die Informationen mit anderen Neuronen austauschen, heißen Interneuronen.

Abb. 2: Nervenzelle

Eine Nervenzelle (s. Abb. 2) besteht aus einem Zellkörper mit Zellkern, vielen astartigen Fortsätzen, die als Eingänge fungieren, einer fadenartigen extrem langen Nervenfaser als Ausgang (im Gehirn etwa einen Millimeter; in Nervensträngen, die Muskeln durchziehen, bis zu einem Meter lang), der an seinem Ende in zahlreiche Nervenendungen oder Endknöpfchen verzweigt. Die Eingänge nennt man Dendriten, den Ausgang Axon; die Verbindung zwischen dem Axon-Ende einer Zelle und dem Dendriten einer anderen Zelle heißt Synapse (des weiteren kennt man Verbindungen Axon-Axon, Dendrit-Dendrit und Axon-Anlagerungen an Dendrit-Dendrit-Kopplungen).

Die Neuronen sind nicht direkt miteinander verbunden, sondern nur mit ihren Ein- und Ausgangsfasern an bestimmten Stellen eng aneinandergepreßt, daher das griechische Wort synapto (eng umgreifend) für diese Kopplung. Zwischen den Membranen (Zellhäuten) der beiden Zellen befindet sich der synaptische Spalt. Eine Information, die von einer zur anderen Zelle übertragen werden soll, muß diesen Spalt überwinden, und das geschieht auf biochemischem Weg, dagegen werden innerhalb einer Zelle Informationen bioelektrisch weitergeleitet. Die Erregung innerhalb der Zellen verläuft in elektrischen Impulsen oder Aktionspotentialen, die, zum Teil nach dem Alles-oder-Nichts-Prinzip erzeugt, ein Signal ohne Abschwächung transportieren; daneben hat man abklingende Übertragung entlang Nervenfasern festgestellt, als kabelartige Verteilung. Ganz grob gesagt, fließt Strom entlang der Zellmembran aufgrund einer Ionenverschiebung, während der Informationsfluß über den synaptischen Spalt mit Hilfe von chemischen Botenstoffen, den Neurotransmittern, sichergestellt wird
In der Überzahl aller Synapsen kooperieren zwei oder drei Botensubstanzen in einer synergetischen (zusammenwirkenden) Art, wodurch in einer Synapse fließende Variationen zwischen den beiden Klassen 'Erregung' und 'Hemmung' vorkommen.

Wie gesagt, Information im Gehirn wird zweifach kodiert: einmal als Impulsfrequenz der Neuronen, eine zeitliche Kodierung, und einmal als Molekülausschüttung der Botensubstanz, eine räumliche Kodierung. Ein Neuron kann ein Signal an viele Nervenzellen weitergeben und von bis zu 200.000 Eingängen Signale aufnehmen; der Durchschnitt liegt bei etwa 10.000. Die Abgabe eines Signals an verschiedene Neuronen wird unter der Funktionsbezeichnung Divergenzprinzip zusammengefaßt. Ein Neuron vervielfacht sein Signal und leitet es an unterschiedliche Stellen weiter. Umgekehrt versteht man unter dem Konvergenzprinzip die Fähigkeit eines Neurons, Signale gleichzeitig aus verschiedensten Bereichen des Gehirns zu empfangen und zu verknüpfen. Bereits die Zusammenschaltung einiger weniger Neuronen zeigt Ansätze kollektiven Verhaltens, das bei größeren Gruppen in eine dynamische sich selbstorganisierende Kommunikation übergeht, die kaum durch das schematisierte 'Lernen' in künstlichen neuralen Netzen beschrieben werden kann. Abgesehen davon ist ein Neuron keineswegs mit einem einfachen Transistor oder ähnlichem zu vergleichen, angebrachter wäre die Analogie zu einem hochintegrierten Chip.

Synapsen übermitteln nicht ausschließlich Signale an das Empfängerneuron, manche von ihnen haben die Aufgabe, Erregungen zu hemmen. Die Tausende von Eingängen eines Neurons enthalten erregende (exzitative) und hemmende (inhibitorische) Synapsen, deren Informationen innerhalb eines Neurons zu einer Gesamtinformation umgerechnet werden. Kompliziert wird die Rechnung, da sie zugleich die Lage der Synapsen quantitativ berücksichtigen muß.

Im Großhirn sitzen viele Synapsen auf dornartigen Auswüchsen, den Spines. Sie sind beweglich, können sich aufzweigen oder zurückbilden und gelten dadurch als verantwortlich für die dynamische Veränderung von Synapsen. Überhaupt: das Charakteristikum der neuralen Vernetzung ist ihre Beweglichkeit und Plastizität.

Gedächtnis bildet sich vermutlich auf der Grundlage dauerhafter Veränderungen von Synapsen-Membranen. Beteiligt an der Informationsspeicherung sind neben Eiweißmolekülen spezielle Verbindungen aus Zukker- und Fettmolekülen. Doch besitzt das Gedächtnis keinen spezifischen Ort, die Inhalte werden über das gesamte Gehirn verteilt abgelegt, aus diesem Grund können Erinnerungen trotz zerstörter Gehirnareale rekonstruiert werden.

Da im menschlichen Gehirn Teile des limbischen Systems sowohl zur Steuerung der Emotionen als auch für das Niederlegen einer Gedächtnisspur verantwortlich sind, werden alle Gedächtnisinhalte mit Gefühlsanteilen abgespeichert und unterliegen dort auch weiterhin einer stimmungshaften Umdeutung.

SALIX PROLOG
FÜR ALLE ATARI ST

STARKE SOFTWARE

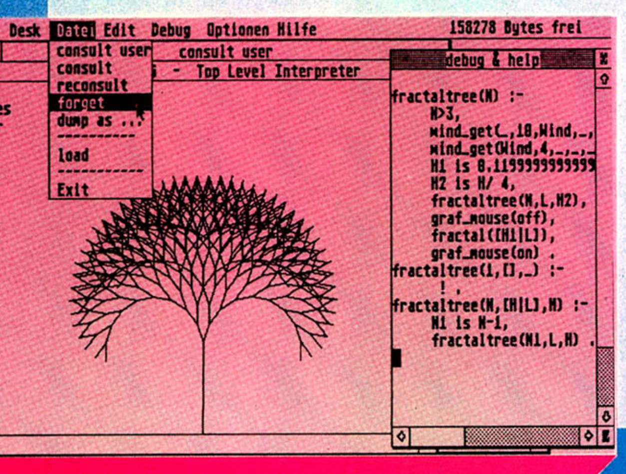
MIT AUSFÜHRLICHEM DEUTSCHEN HANDBUCH

Leistungsdaten:

- ▶ **Compiler/Interpreter System.** Der inkrementelle Compiler erzeugt einen Zwischencode, der vom Interpreter abgearbeitet wird. Geschwindigkeit 1200 Lips.
- ▶ **Edinburgh Standard.** Alle im Buch 'PROLOG – Programming in Logic' von Clocksin & Mellish definierten Funktionen sind verfügbar.
- ▶ **Zusätzliche eingebaute Funktionen,** (primitives), insgesamt ca. 160 Funktionen
- ▶ **Integer- und Gleitkommaarithmetik,** mathematische Funktionen, Gleitkomma in doppelter Genauigkeit.
- ▶ **Syntax Error Corrector.** Tritt beim Einlesen eines PROLOG-Ausdrucks (vom Bildschirm oder vom Laufwerk) ein Syntaxfehler auf, erhält der Benutzer Gelegenheit, den Fehler am Bildschirm zu verbessern.
- ▶ **Lector.** Ein in PROLOG geschriebenes Hilfsmittel, um PROLOG-Programme auf Tippfehler und falsche Verwendung der eingebauten Funktionen zu prüfen.
- ▶ **Anschluß an einen vorhanden Editor.** (Z. B. 1st-Word) Die PROLOG-Sitzung muß nicht verlassen werden, um eine Bibliothek zu editieren.
- ▶ **GEM-Library** gestattet die Nutzung fast aller GEM-Funktionen von PROLOG aus, damit steht die riesige Grafikbibliothek des GEM in der Form von PROLOG-Prädikaten zur Verfügung.
  Zusätzlich Window Management System mit Pufferung und automatischem Refresh der Fenstertextinhalte.
- ▶ **GEM-Top-Level-Interpreter.** Komfortable Bedienung des PROLOG-Programmiersystems mit Drop-Down-Menüs und Formularen. Der Gem-Top Level Interpreter ist komplett in PROLOG geschrieben, damit auch für spezielle Anforderungen modifizierbar.
- ▶ **Startup-File** um das PROLOG-System entsprechend den jeweiligen Anforderungen zu konfigurieren.
- ▶ Der **volle Adressraum des 68000** (auch 4 MB) wird unterstützt, so volle Ausnutzung des ATARI-Speichers und etwaiger Speichererweiterungen.
- ▶ **Deutsches oder englisches Handbuch** mit mehr als 300 Seiten
- ▶ **Datenbankeditor**
- ▶ **Zyklische Strukturen** werden verarbeitet
- ▶ **Exception Handling**
- ▶ **Leistungsfähiges Testsystem**

- ▶ **Benutzerdefinierte Funktionen**
- ▶ **Globale Variable**
- ▶ **Clipboard-Device**

DM 198,-



Handbuch einzeln **DM 60,-** (Wird beim Kauf angerechnet)

DIE SALIX PROLOG SYSTEMFAMILIE BEIM HEIM-VERLAG

NEU: SALIX PROLOG V2.8

▶ **noch schnell durch** clause-indexing, structure - controlled garbage collection
▶ **noch mehr Funktionen** - einstellbares Gleitkommaformat - Function Key support - GEM-Formular als Ausgabemedium
▶ **noch mehr Komfort** - Top-Level-Interpreter konfigurierbar - programmierbare Funktionstasten - History-Funktion
▶ **noch mehr Sicherheit** - eigene Exceptionbehandlung

Update: DM 39,-
Preis: DM 198,-

NEU: PROLOG_for_C

Die Neuigkeit für den C-Programmierer: Der SALIX-PROLOG Interpreter läßt sich nun als Unterprogramm innerhalb eines C-Programmes verwenden. Damit erschließt sich die Welt des „Programmierens in Logik" auch konventionellen C-Programmen und das zu einem sensationellen Preis. Wir liefern:

▶ **SALIX PROLOG Interpreter V2.8**
▶ **C-Interface**
▶ **einfacher in C geschriebener TOP-Level-Interpreter als Demonstrationsprogramm**
▶ **PROLOG Handbuch**

Preis: DM 99,- **Preis ohne Handbuch: DM 59,-**

NEU: SALIX PROLOG Professionell V2.8

wie SALIX PROLOG Basisversion plus:
▶ **Modulkonzept** - Definition von abgeschlossenen Modulen mit verschiedenen Namens- bzw. Prädikatenräumen vermeidet die Gefahr von Namenskonflikten und erlaubt Teamarbeit bei größeren Projekten. Schnelladen fertiger Module.
▶ **C-/Assemblerschnittstelle** - C- und Assemblerprogramme können als benutzerdefinierte Primitives definiert und dynamisch eingebunden werden. Beispielprogramme sind enthalten.
▶ **Prolog_for_C ist enthalten**
▶ **TOS- und GEM-Oberfläche**
▶ **kostenlose Runtime-Lizensen**

Preis: DM 498,- **Upgrade von Basisversion: DM 300,-**

NEU: SALIX DIALOGEXPERTE V1.2

enthält Anpassungen des Dialogexperten an PROLOG V2.8 plus einige Erweiterungen in Form zusätzlicher GEM-High-Level-Prädikate, insbesondere Function Key Support.

Update: DM 29,-
Preis: DM 79,-

Alle Preise sind unverbindlich empfohlene Verkaufspreise.

Schweiz
DataTrade AG
Langstr. 94
CH - 8021 Zürich

Österreich
Haider
Computer + Peripherie
Granzer Str. 63
A-2700 Wiener Neustadt

Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 0 61 51 - 5 60 57

BESTELL-COUPON an Heim-Verlag, Heidelberger Landstraße 194, 6100 Darmstadt-Eberstadt

Bitte senden Sie mir: ____________
zzgl. DM 5,- Versandkosten (unabhängig von bestellter Stückzahl)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei
Name, Vorname ____________
Straße, Hausnr. ____________
PLZ, Ort ____________
Benutzen Sie auch die in ST COMPUTER vorhandene Bestellkarte.

ne bzw. Programm und Gehirn glaubhaft machen. Man wird sehen: Die Programmierung mit Hilfe neuraler Netze stellt eine interessante Neuerung dar, bloß der tatsächlichen Informationsübertragung und -verarbeitung des Gehirns wird sie bei weitem nicht gerecht. Warum nicht, das soll in den folgenden Abschnitten erläutert werden.

Neurale Landschaft

Um die regulierte Entwicklung eines Embryos aus einer befruchteten Eizelle zu verdeutlichen, hat der britische Biologe Conrad Waddington das eindrucksvolle Bild von der "epigenetischen Landschaft" gewählt. Gemeint ist damit folgendes: Während der Embryo zeitlich die Topologie der epigenetischen Landschaft, seine ihn formende Umwelt also, durchquert, kanalisiert die Interaktion mit chemischen und physikalischen Umgebungen die werdenden Einzelorgane in verschiedene Entwicklungsbahnen und verleiht zugleich dem sich entfaltenden Organismus insgesamt Gestalt.

Bildhaft stelle man sich diese Landschaft als ein von zahlreichen Erhöhungen und Vertiefungen geprägtes Gelände vor. Wird dort eine Kugel auf einen Hügel gelegt, beginnt sie abwärts zu rollen, die anliegenden Täler (Entwicklungspfade) hinab, in Abzweigungen hinein, bis sie endlich am Grunde zur Ruhe kommt. Die Kugel steht hier als Symbol für ein sich selbst organisierendes System (zum Beispiel eine Eizelle), das in festgelegten Rahmenbedingungen (so können Höhen biologisch-genetisch nicht erreichbare Zustände charakterisieren) automatisch, ohne den Weg zu kennen, seinen stabilen Zielzustand findet.Genau diese Selbstorganisationsprinzipien glaubt man in den neuralen Geflechten des Gehirns auszumachen. Die Synapsen, die Verbindungen zwischen einzelnen Nervenzellen, übertragen Botschaften, die sie nicht kennen. Trotzdem funktionieren sie zielgerecht, indem sie bestimmte stabile (damit informationsfördernde) Ordnungszustände (Täler) einnehmen.

Hopfield ersetzt das Bild der epigenetischen Landschaft durch die Energielandschaft eines speziellen Schaltkreises, dem Flip-Flop (8). Die zentralen Einheiten der Schaltung bilden zwei Verstärker, die dadurch gekennzeichnet sind, daß ihre Ausgangsspannung mit wachsender Eingangsspannung zunimmt, bis ein Maximum erreicht wird, ab dem es trotz weiterer Erhöhung der Eingangsspannung keinen Zuwachs mehr auf der Ausgangsseite gibt. Auch die umgekehrte Konstellation ist gültig: Eine fallende Eingangsspannung führt zu einem Abfall der Ausgangsspannung bis hin zu einem Minimum. Invertiert man die Ausgangsspannung eines jeden Verstärkers und führt sie dem Eingang des jeweiligen anderen zu, erhält man ein Flip-Flop, das seinem lautmalerischen Namen entspricht. Es rastet sofort in einen von zwei stabilen Zuständen ein, d. h. entweder fällt bei einer hohen Ausgangsspannung von Verstärker A die Ausgangsspannung bei B ins Minimum oder eine hohe Ausgangsspannung von B veranlaßt am Ausgang von A einen Tiefpunkt.

Gleich mit welcher Anfangsspannung man das Flip-Flop einschaltet, alsbald verharrt es in einem der beiden stabilen Zustände. Die Fallkurve der Energie, ausgehend von beliebigen Initialpunkten und endend in den Minima, kann man berechnen und grafisch als Landschaftsrelief darstellen (s. Abb. 1). Kennzeichnet man Maximum und Minimum am Ausgang eines Verstärkers mit +1 bzw. -1 und versetzt Verstärker A in den +1-Zustand, sinkt die Spannung am Eingang von B, zugleich kippt der Ausgang von B in den -1-Zustand. Die invertierte Ausgangsspannung von B wiederum stabilisiert den Ausgang von A bei +1. Entgegengesetzt verläuft die Entwicklung aus der Ausgangssituation -1 bei A. Die beiden Achsen des unterteilten Rechtecks in Abb. 1 repräsentieren die Ausgangsspannung eines jeden Verstärkers, die dritte (vertikale) Achse zeigt den Wert der berechenbaren Energie an. Täler (stabile Zustände) bilden sich bei den beiden Konfigurationen -1,+1 und +1,-1.

Verformen läßt sich die Oberfläche der Landschaft durch Auswahl und Variation der Bestandteile der Schaltung (Art der Verstärker, Spannungspegel, zwischengeschaltete Widerstände, zugeführte Stromquellen etc.). Wird beispielsweise dem Eingang eines Verstärkers eine positive Stromquelle zugeführt, hat dies eine Vertiefung des Tales auf seiner Seite zur Folge. Die Form der Landschaft entwickelt sich somit in einem kollektiven Zusammenwirken aller Bestandteile der Schaltung.

Bereits die einfache Konfiguration aus zwei Flip-Flops kann schlichte Entscheidungen treffen, etwa die, welche von zwei gegebenen Zahlen die größere ist. Führt man unter dieser Fragestellung den Eingängen der Verstärker externe Stromquellen zu, deren Spannung proportional den zu untersuchenden Zahlen gewählt wird, erzeugt der Verstärker mit der höheren Eingangsspannung ein tieferes Energietal an seinem +1-Zustand. Nach Stabilisierung der Schaltung symbolisiert der Verstärker, der sich im +1-Zustand befindet, die größere Zahl.

Wirklich interessant werden erst vielfache Zusammenschaltungen von Flip-Flops, mit denen sich dann anspruchsvollere Aufgaben lösen lassen. Hopfield und Tank zeigen, wie eine kollektive n-Flop-Schaltung ein sogenanntes task-assignment-Problem (die Zuordnung verschiedener Aufgaben an Personen, die diese Arbeiten in unterschiedlicher Geschwindigkeit ausführen) in allerkürzester Zeit bewältigen kann.

Lernen im neuralen Netz

Vor allem die Vorteile der natürlichen Informationsverarbeitung, also die breite Aufnahme und Bewertung gleichzeitig einströmender Informationen, die Speicherung von Daten über das gesamte Medium und nicht zuletzt die Fähigkeit, ohne besondere Instruktionen zu lernen, sollten gleichermaßen die künstlichen neuralen Netze auszeichnen. Dementsprechend basieren solche Systeme prinzipiell auf paralleler Verarbeitung, systemverteiltem 'Wissen' und sich anpassendem Lernen.

Wie schon beim Perceptron baut man ein idealisiertes Neuronennetz mit Einfachst-Versionen von Neuron, Synapse und neuralen Verbindungen auf. Als Gewinn winken theoretisch fundierte und mathematisch beschreibbare Abläufe des Informationsflusses mit dazugehörigen Regeln zur Speicherung und Rückgewinnung der Informationen.

Sehen wir uns die am häufigsten eingesetzte künstliche Netzwerkstruktur an (10). Idealerweise sollte sie aus vielen bis sehr vielen gleichartigen Prozessoren (Neuronen) bestehen, die untereinander verbunden sind (Dendriten und Axone) und miteinander kommunizieren können (Synapsen). Der Extremfall einer vollständigen Verknüpfung (jede Einheit ist mit jeder anderen verbunden) ist dabei nicht gefordert, vielmehr wird ein Schichtenaufbau eingesetzt, in welchem Informationen in einer Richtung, vorwärts vom Systemeingang zum Systemausgang, durchgeschleust werden (feed-forward-Netz).

Schematisch dargestellt, sieht ein solches neurales Netz wie in Abb. 3 aus. Jeder Knoten des Netzes verkörpert ein Neuron und läßt sich durch eine Zahl, seine Aktivierung, kennzeichnen; ebenso wird jede Verbindung durch eine Gewichtszahl dargestellt. Das Eingangsmuster wird von einer Schicht Eingangsneuronen aufgenommen, die in der Regel verknüpft sind mit internen ('verborgenen') Schichten von Neuronen. Diese erhalten als Eingang die gewichtete Summe der Ausgänge der vor ihr liegenden Schicht. Entsprechend dem Grad ihrer Aktivierung geben die Neuronen Ausgangssignale an die nächste Schicht weiter, bis schließlich in der letzten, der Ausgangsschicht, das Zielmuster erscheint.

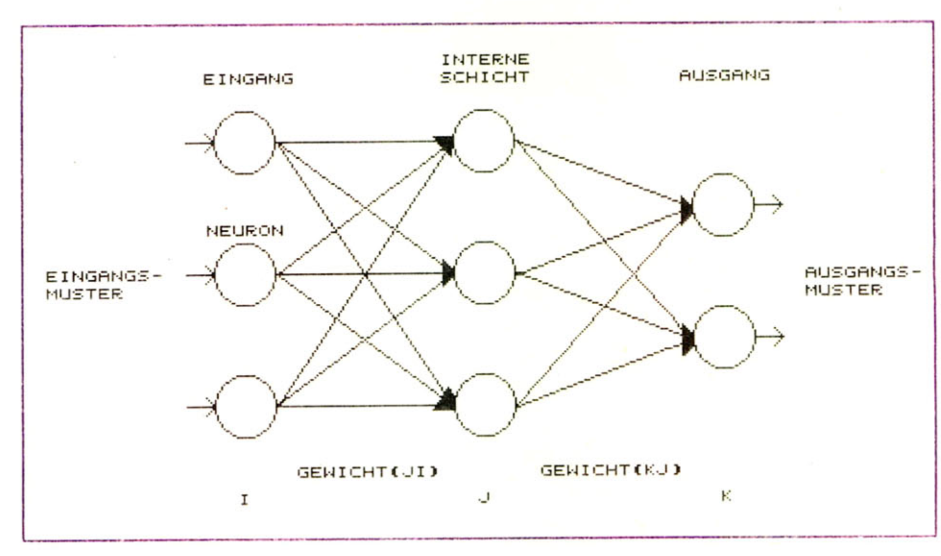


Abb. 3: Neurales Netz mit Vorwärtsausbreitung

Mathematisch läßt sich der Netzeingang (alle Neuronen der Eingangsschicht *i* umfassend) bei einem Neuron der ersten internen Schicht *j* in der Formel

$$\text{Ausgangswert}_i = \text{Eingangswert}_i$$
$$\text{Netzeingang}_j = \text{Summe}(\text{Gewichtswert}_{ji} * \text{Ausgangswert}_i)$$

fassen. Das heißt: jedes Neuron der ersten internen Schicht ist mit allen Neuronen der Eingangsschicht verbunden, wobei letztere als Ausgangswert ihren Eingangswert abgeben, multipliziert mit einer jeweils auf das nachgeschaltete Neuron bezogenen Gewichtszahl; insgesamt erhält das nachgeschaltete Neuron somit die gewichtete Summe aller Eingangswerte. Jedes Neuron der ersten internen Schicht verarbeitet nun die gewichtete Summe der Eingangssignale und gibt seinerseits ein Ausgangssignal ab, das wiederum mit Gewichtungen belegt, an alle Neuronen der nächsten Schicht gesandt wird. Während jedes Neuron seine spezifischen Gewichtszahlen besitzt, ist die Verarbeitungs- oder Aktivierungsfunktion für alle Neuronen gleich.

Zur Aktivierung greift man auf eine sigmoide (s-förmige) Funktion zurück, die Werte zwischen 0 und +1 ausgibt und unter bestimmten Bedingungen Schwellenwert-Eigenschaften aufweist:

$$\text{Aktivierungswert}_j = 1 / (1 + \text{EXP}(-(\text{Netzeingang}_j + \text{Schwelle}_j) / \text{Steilheit}_j))$$

Strebt die Steilheit, die den Grad des Anstiegs der Funktionskurve bestimmt, gegen 0, nähert sich die Funktion immer mehr einem Schwellenwert-Element (Treppenfunktion) und bringt bis auf einen kleinen Übergangsbereich entweder die Werte +1 oder 0 zur Ausgabe. Der Parameter Schwelle verschiebt die Funktionskurve entlang der Netzeingangs-Achse.

Der Netzeingang eines Neurons der nächsten Schicht k besteht nunmehr aus der gewichteten Summe der Aktivierungswerte der vor ihr liegenden Neuronen der Schicht *j*:

$$\text{Ausgangswert}_j = \text{Aktivierungswert}_j$$
$$\text{Netzeingangk} = \text{Summe}(\text{Gewichtswertk}_j * \text{Ausgangswert}_j)$$

Schicht um Schicht werden die jeweils aktivierten und gewichteten Werte verbreitet (propagiert), bis sie an den Ausgangsneuronen das Zielmuster abbilden.

Nur in einfachen, aus wenigen Schichten bestehenden Netzen kann man noch ohne großen Aufwand die Gewichte und Schwellenwerte 'per Hand' festlegen. So löst ein Netzwerk aus zwei Eingangsneuronen und einem Ausgangsneuron mit unterschiedlichem Schwellenwert die Boolesche AND- und OR-Funktion (s. Abb. 4). Übertrifft die gewichtete Summe der Eingabewerte den Schwellenwert des Ausgangsneurons, erhält die Ausgabe den Wert +1, sonst 0.

Doch gerade die Erstellung der Gewichte und Schwellenwerte sollten neurale Netze selber lösen. Daher nehmen diese Methoden breiten Raum innerhalb des Funktionsablaufs ein.

Prinzipiell gibt man dem Netz in einer Lernphase ein Eingabemuster vor, ordnet den Gewichtsvariablen zufällige Werte zu und schleust die in jeder Schicht neu verarbeiteten Werte bis zum Ausgabemuster durch das Netz. Dann vergleicht man das errechnete Ausgabemuster mit dem gewünschten Zielmuster und korrigiert rückwärts (vom Ausgang zum Eingang) die Gewichte im Hinblick auf eine Minimierung der Fehler (Back-Propagation of Error). Mit den justierten Gewichten kann ein weiteres Eingabemuster verarbeitet und der gesamte Prozeß erneut durchlaufen werden.

Eine gebräuchliche Lernfunktion stellt die verallgemeinerte Delta-Regel dar (1986 von David E. Rumelhart, Geoffrey E. Hinton und R. J. Williams veröffentlicht):

$$\text{DELTA}(\text{Gewichtswert}_{ji}(_{n+1})) =$$
$$\text{eta} * \text{delta}_j * \text{Ausgangswert}_i + \text{alpha} * \text{DELTA}(\text{Gewichtswert}_{ji}(_n))$$

Mit $(_{n+1})$ bzw. $(_n)$ wird der Iterationsschritt angegeben; die Konstante *eta* bezeichnet den Lernfaktor; *alpha* ist eine Momentum-Konstante zur Dämpfung der Oszillationen. Die Fehlerkorrekturvariable delta enthält die Differenz zwischen erwünschter (Zielwert) und tatsächlicher (Ausgangswert) Ausgabe. Im einzelnen berechnet man für *DELTA* zuerst die *delta*-Korrektur der Ausgangsschicht *k* einer Vorgabe *v*:

$$\text{delta}_{vk} = (\text{Zielwert}_{vk} - \text{Ausgangswert}_{vk}) * (1 - \text{Ausgangswert}_{vk}) * \text{Ausgangswert}_{vk}$$

Darauf folgt die Berechnung der *delta*-Korrektur der internen Schicht *j*, die sich gewinnen läßt mit Hilfe der Formel:

$$\text{deltavj} = \text{Summe}(\text{delta}_{vk} * \text{Gewichtswert}_{kj}) * (1 - \text{Ausgangswert}_{vj}) * \text{Ausgangswert}_{vj}$$

Mit den jeweiligen *DELTA*-Werten werden alle Gewichte eines Vorgabemusters justiert.

$$\text{Gewichtswert}_{ji} = \text{Gewichtswert}_{ji} + \text{DELTA}(\text{Gewichtswert}_{ji})$$

Im Anschluß daran wird die Fehlersumme der Vorgabemuster:

$$\text{Fehlersumme} = \text{Summe}((\text{Ziel}_{vk} - \text{Ausgang}_{vk})^2)$$

und der Systemfehler über alle Muster erstellt:

Systemfehler =
0,5 * Fehlersumme / Anzahl der Muster

Erst wenn der Systemfehler ein Toleranzminimum unterboten hat oder die vorgegebene maximale Anzahl von Iterationen überschritten wurde, bricht die Lernphase ab, im ersten Fall erfolgreich, im zweiten erfolglos. Im erfolgreichen Fall hat das Netz gelernt, Eingabemuster mit gewünschten Ausgabemustern derart zu verknüpfen, daß es auch auf 'unscharfe' neue Eingaben mit den 'richtigen' Ausgaben reagiert.

Im Listing sind in den DATA-Anweisungen die XOR-Bedingungen abgelegt: Die beiden jeweils ersten Zahlen geben die Eingaben wieder, die abgesetzte dritte Zahl die gewünschte Ausgabe, die wegen der in der Aktivierungsfunktion auf 1 gesetzten *Steilheit* statt 1 den Wert .9 und statt 0 den Wert .1 annehmen muß - andernfalls verlangt die Formel unendlich große positive oder negative Netzeingänge. Im übrigen werden im Programm die Werte für den Parameter Schwelle in der gleichen Weise wie für die Gewichtszahlen ermittelt.

Das XOR-Problem entspricht dem 'Parity-2-Problem', d. h. immer wenn die Summe der Eingangswerte eine gerade Zahl ergibt, wird der zugehörige Ausgangswert auf 1 gesetzt. Ein neurales Netz ohne interne Schichten (wie das Perceptron) versagt bei dieser Problemstellung. Der Verlauf der Lernkurve für das XOR-Problem läßt sich in etwa folgendermaßen charakterisieren: Beginnend bei einem Systemfehler von 0,10, fällt die Lernkurve nach wenigen Iterationen auf 0,08, sinkt im Verlauf der nächsten zwei- bis dreihundert Iterationsschritte auf 0,05, danach vermindert sich der Systemfehler schneller, erreicht aber frühestens nach über 400 Durchläufen die untere Fehlergrenze.

Anzumerken sei noch, daß nicht mit gleichen Gewichtswerten begonnen werden darf und daß unter Umständen das Netz in lokalen Minima hängen bleiben kann; oftmals hilft ein Neustart.

Selbstverständlich hat man sich im Rahmen des Konnektionismus auch an den Nachbau eines Gedächtnisses gewagt. Nachzuahmen versucht man hier die Fähigkeit des menschlichen Gedächtnisses, aus unvollständigen Vorgaben die zugehörigen komplexen Gesamteindrücke ins Bewußtsein zu rufen. Beispielsweise kann eine Melodie, ein Geruch, eine bestimmte Atmosphäre, manchmal ein bloßes Wort einen ganzen Erinnerungsfilm ablaufen lassen. Ein beeindruckendes Vermögen unseres Gedächtnisses, das schon Aristoteles fasziniert und veranlaßt hat, eine Schrift über die unwillkürliche Erinnerung mit ihrer außerordentlichen Fähigkeit zur Assoziation (Verknüpfung von Vorstellungen) zu verfassen.

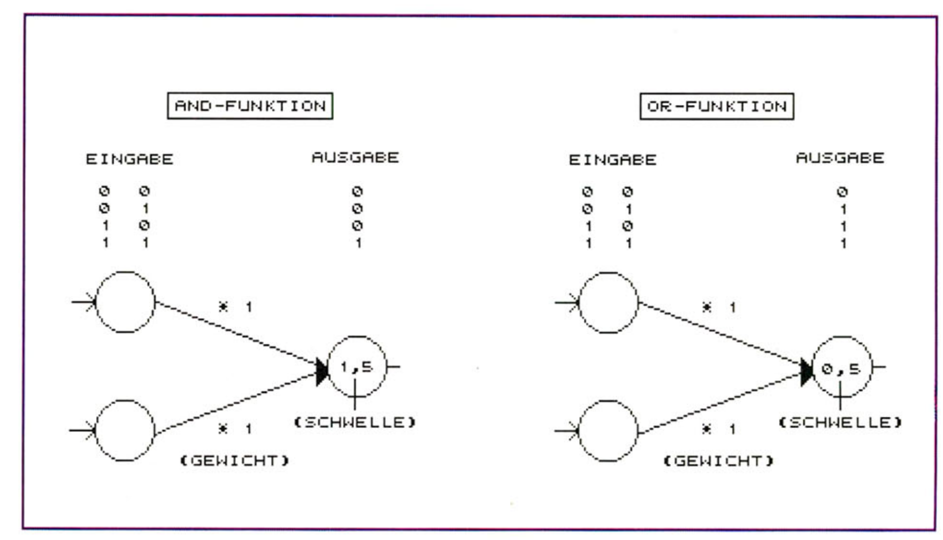


Abb. 4: Neurales Netz für AND- und OR-Funktion

Gewöhnlich wird diese Fähigkeit in konnektionistischen Modellen reduziert auf eine mathematische Verknüpfung aller Eingabeelemente und die Abspeicherung der Verknüpfungszahlen in Form einer Gedächtnismatrix, aus der später mit einer Teilabfrage die vollständige Information wieder ausgelesen werden kann. Im Gegensatz zur verallgemeinerten Delta-Regel liegen zu verschiedenen Versionen von Matrix-Speichern Programme in deutschsprachigen Publikationen vor, insofern möchte ich hier nur auf die entsprechenden Artikel verweisen (11).

Zum Stand der Dinge

Glaubt man manchen Propheten des Konnektionismus, erwarten uns wieder - wie bei der klassischen KI - schöne Aussichten auf Maschinenintelligenz, besonders dann, wenn neurale Netze in Hardware-Schaltungen vorliegen. Zwar werden gegenwärtig noch neun Zehntel aller neuralen Netzwerke auf sequentiellen Computern simuliert, mit einer in der Regel quälend langsam ablaufenden Lernphase. Gedacht und konzipiert worden sind die Modelle dagegen für, wie es heißt, massive Parallelverarbeitung. Wie sieht es denn nun aus mit dem Stand der Forschung und den Anwendungsgebieten?

Zunächst ein Blick auf die Bundesrepublik. Ein Verbundvorhaben des Bundesministers für Forschung und Technologie zur "Informationsverarbeitung in neuronaler Architektur" setzt drei Anwendungsgebiete in den Mittelpunkt der Forschung: a) die Verarbeitung visueller Informationen, b) die Motorsteuerung und c) die Spracherkennung (12). Noch befindet sich alles in der Experimentierphase, mit allerdings sechs Millionen DM Förderungsgeldern für das Jahr 1989.

Weiter sind die US-Amerikaner, auch in ihren Ansprüchen. "Wir versuchen, ein Nervensystem auf Silizium-Basis aufzubauen", hört man von Carver Mead (13). Immerhin hat er eine künstliche Retina aus 100.000 Transistoren konstruiert, die besser arbeiten soll als jedes andere von Computern gesteuerte visuelle System. Seit längerem kennt man NETtalk von Charles Rosenberg und Terrence J. Sejnowski. Ihr Computerprogramm, das eine Zeichenkette in Phoneme umwandelt, läuft auf einer VAX 11/780 und stellt ein neurales Netz mit drei Schichten dar. Auf der Eingangsseite befinden sich 7 * 29 Neuronen, mit denen gleichzeitig 7 Buchstaben (einschließlich Satz- und Leerzeichen) erfaßt werden können. Eine verdeckte Schicht verarbeitet die gewichteten Informationen und leitet das Ergebnis den 55 Neuronen der Ausgangsseite zu.

Ein neurales Netz besonderer Art hat Demetri Psaltis entwickelt. Es besteht nicht wie üblicherweise aus elektronischen Schaltungen, sondern aus einer netzartigen Anordnung von Lasern, Linsen, Spiegeln und Hologrammen. In den Hologrammen sind die Gesichter ver-

schiedener Personen gespeichert, zum Abruf benötigt das Netz nur den Ausschnitt eines Bildes. Ein Ergebnis, das auch Teuvo Kohonen mit seinem assoziativen Gedächtnis erreicht. Robert Hecht und Todd Gutschow haben auf einer Zusatzkarte zu Personal-Computern 30.000 Neuronen mit 480.000 Verbindungen untergebracht. Die Reihe ließe sich fortsetzen: die große Bewährungsprobe in der praktischen Anwendung jedoch fehlt noch.

Daran ändert auch die nachfolgende Einschätzung nichts, in der zudem eine gewisse Ironie liegt. Expertensysteme haben das Perceptron verdrängt und werden jetzt zum Teil auf ihrem eigenen Gebiet von mehrschichtigen perceptronähnlichen neuralen Netze überholt, wie das Beispiel von AMPS, einem experimentellen akustischen Motorenprüfsystem, zeigt. Im Vergleich mit einem konventionellen Expertensystem übertraf es dies nach Begutachtung menschlicher Experten um den Faktor 1,6 bis 6,0, bei nur zwei 'Mannwochen' Implementierungs- und Lernzeit (14).

Eines ist sicher, künstliche neurale Netze sind eine Bereicherung der Softwareszene, ob sie jedoch in ihrer Mehrzahl bessere Lösungen bieten als herkömmliche Programme, ist zweifelhaft. So gesehen ist die Einschätzung Marvin Minskys, eines der Väter der klassischen KI, nur konsequent: "Wenn sie (die neuralen Netze) irgendetwas leisten, das auf anderem Wege nicht getan werden kann, *das* wäre revolutionär." (15)

Aber auch dann noch trennten sie Welten vom natürlichen Nervensystem, über das man so viel und zugleich so wenig weiß. Befragt, wie denn die Riesenkluft zwischen neuralem Geschehen und Bewußtsein mit ein paar simplen elektronischen Schaltungen zu überspannen sei, antwortet die Gehirnforscherin und Philosophin Christine Skarda von der Universität Berkley offenherzig: "Nobody really has any idea. It's just a wonder tissue." (16)

Dr. A Ebeling

Literatur:

(1) Vollmer, G.: "Evolutionäre Erkenntnistheorie", Hirzel, Stuttgart 1987, S. 86

(2) Jursa, O: "Kybernetik, die uns angeht", Bertelsmann, Gütersloh, S. 277

(3) Möcke, F.: "Drei Gutachten zu Anwendungen und Folge von Expertensystemen", in: c't 1989/7, S. 25

(4) Redaktion der Time-Life-Bücher: "Künstliche Intelligenz", 1987, S. 13

(5) Vgl. (1), S. 42 f. und 124 f.

(6) Ebeling, A.: "Gehirn, Sprache und Computer", Heise, Hannover 1988, S. 83

(7) Mechsner, F.: "In den Gespinsten des Geistes", in: Geo-Wissen "Gehirn-Gefühl-Gedanken", 1987, S. 136

(8) Tank, D. W; Hopfield, J. J.: "Collective Computation in Neuronlike Circuits", in: Scientific American, Special Issue: "Trends in Computing", 1988, S. 54-61

(9) Nach (6), S. 40 ff.

(10) Vgl. Pao, Y.-H.: "Adaptive Pattern Recognition and Neural Networks", Addison-Wesley, Reading 1989, S. 113 ff.

(11) Zu erwähnen seien hier die Artikel von Sven B. Schreiber in der "c't" (1987/4, S. 98-101; 1987/7, S. 106-114; 1988/7, S. 114-125) zum gegengekoppelten Matrixgedächtnis, das von Prof. Teuvo Kohonen in Finnland konzipiert wurde. Programme zum Matrix-Gedächtnis ohne Rückkopplung und zum 'Traveling-Salesman-Problem', entwickelt von einer Arbeitsgruppe der TU-München unter Leitung von Prof. Klaus Schulten, findet man in der "mc" (1987/9, S. 108-120; 1989/2, S. 48-61). Von Joachim Bentz stammt ein Matrixgedächtnis als assoziativer Speicher, abgedruckt in der "c't" (1988/10, S. 84-92). Im 7. Teil seiner Einführung in die KI stellt Karl Sarnow im "ST-Computer" (1987/11, S. 124-128) ein Programm zum Matrix-Gedächtnis nach einem Prinzip von John Hopfield vor. E.-D. Schmitter beschreibt in "elcomp" (1989/5-6, S. 19- 30; 1989/7-8, S. 21-31) neurale Netze, ohne allerdings eine explizite Beschreibung der Delta-Regel zu liefern.

(12) Informationspapier des Bundesministers für Forschung und Technologie zur "Informationsverarbeitung in neuronaler Architektur", 1988

(13) Heppenheimer, T. A.: "Nerves of Silicon", in: Discover 1988/2, S. 79

(14) Ratajczak, P.: "Workshop: Neuronale Netze", in c't 1989/8, S. 18

(15) Siehe (13), S. 76

(16) Allman, W. F.: "Mindworks", in: Science 1986/5, S. 31

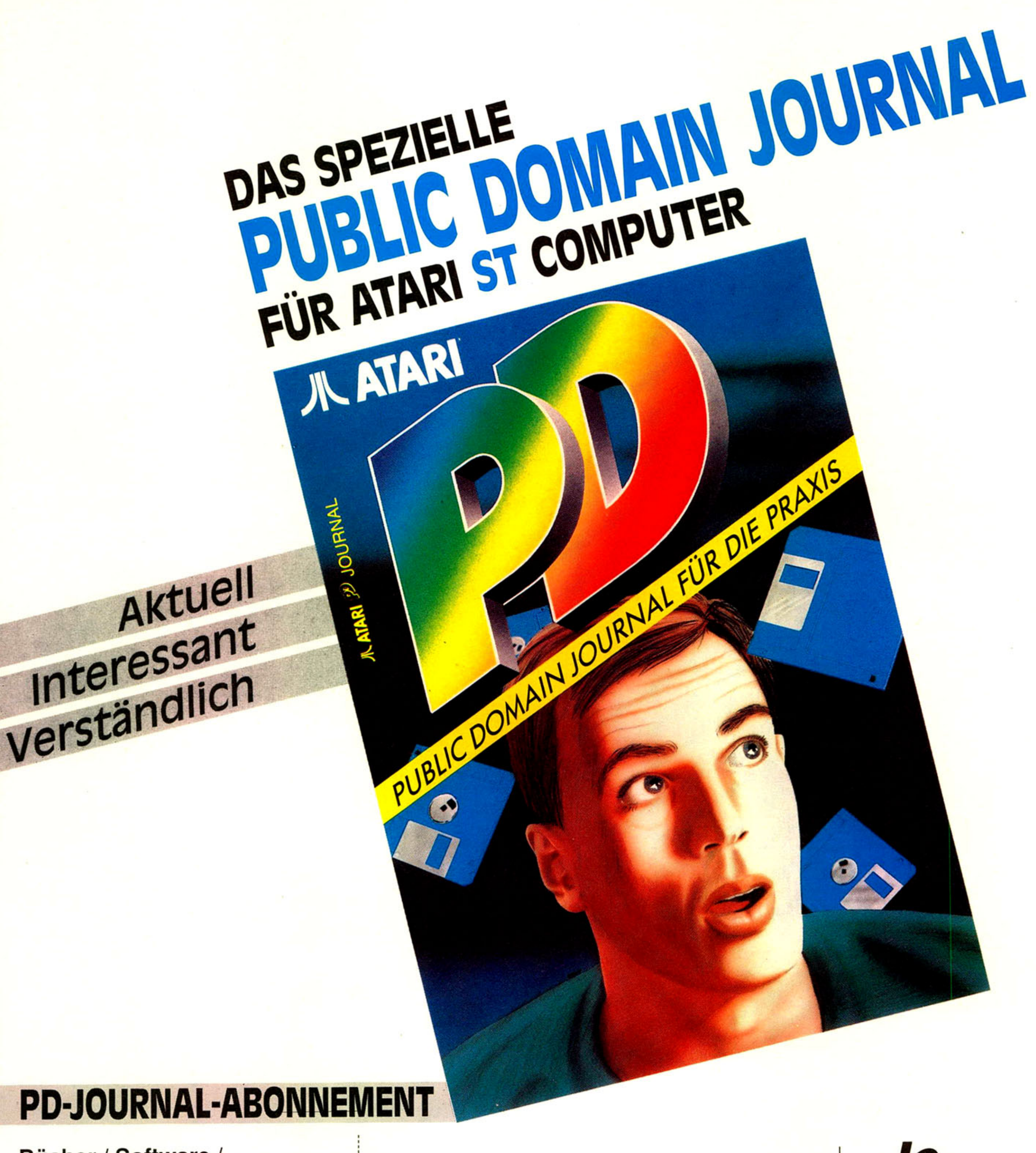

```
' Neurales Netz mit verallgemeinerter Delta-Regel
  (GFA-BASIC Vs. 3)
' nach Pao, Y.-H.: Adaptive Pattern Recognition
  and Neural Networks
' (c) MAXON Computer GmbH
'
' Konstanten deklarieren
'
max_muster%=100                ! max. Anzahl
                                 Eingangsmuster
max_eingangs_neuronen%=50      ! max. Anzahl
                                 Eingangsneuronen
max_schichten%=4               ! max. Anzahl
                                 verdeckter Schichten
max_neuron%=50                 ! max. Anzahl Neuronen
                                 in verdeckter Schicht
max_ausgangs_neuronen%=50      ! max. Anzahl
                                 Ausgangsneuronen
max_durchgang%=3000            ! Anzahl der
                                 Iterationen
eta#=0.9                       ! Lernfaktor
alpha#=0.7                     ! Stabilisator
weiter%=0                      ! Abbruch-Konstante
erfolglos%=1                   ! Abbruch-Konstante
erfolgreich%=2                 ! Abbruch-Konstante
'
' Dimensionieren
'
DIM gewicht#(max_schichten%+1,max_neuron%)
DIM ausgang#(max_schichten%+2,max_neuron%)
DIM delta#(max_schichten%+2,max_neuron%)
DIM delta_gewicht#(max_schichten%+1,max_neuron%)
DIM ziel#(max_muster%,max_ausgangs_neuronen%)
DIM eingang#(max_muster%,max_eingangs_neuronen%)
DIM fehler#(max_muster%)
DIM ausgabe#(max_muster%,max_ausgangs_neuronen%)
DIM neuron_pro_schicht%(max_schichten%+2)
'
' --------------- Hauptprogramm -------------
'
voreinstellen
initialisieren
daten_einlesen
PRINT AT(25,1);"Neurales Netz"
PRINT AT(5,2);"Lernen nach der verallgemeinerten
               Delta-Regel"
PRINT
DO
  system_fehler#=0
  FOR i%=0 TO muster%-1
    vorwaerts_propagieren(i%)
    delta_ausgang_berechnen(i%)
    delta_vschicht_und_eingang_berechnen
    gewichte_aendern
    fehler_berechnen(i%)
  NEXT i%
  system_fehler#=0.5*system_fehler#/muster%
  INC durchgang%
  pruefen%=FN abbruch
LOOP WHILE (pruefen%=weiter%)
gewichte_und_schwellen_ausgeben
IF pruefen%=erfolglos%
  PRINT
  PRINT "max. Anzahl der Durchlaeufe erreicht"
  PRINT "Minimierung fehlerhaft"
  PRINT "Ende = <Taste>"
  VOID INP(2)
ELSE
  gelerntes_abfragen
ENDIF
```

→

```
END
' ---------------------------------------------
'
'   Beispiel: XOR-Problem
'   ---------------------
'   Einga- Einga-  Ausgabe
'   be 1   be 2
'
DATA 0,      0,      .9
DATA 0,      1,      .1
DATA 1,      0,      .1
DATA 1,      1,      .9
'
PROCEDURE voreinstellen
  LOCAL i%
  ' Voreinstellung
  muster%=4
  eingangsneuronen%=2
  ausgangsneuronen%=1
  min_systemfehler#=0.0001
  v_schicht%=1
  neuron_pro_schicht%(1)=2
  '  PRINT "Systemeingaben:"
  ' INPUT "Anzahl der Eingabemuster (100):",
          muster%
  ' INPUT "Anzahl der Eingangsneuronen (2-50):",
          eingangsneuronen%
  ' INPUT "Anzahl der Ausgangsneuronen (1-50):",
          ausgangsneuronen%
  ' INPUT "minimaler Systemfehler (0.01 -
          0.0001): ",min_systemfehler#
  ' INPUT "Anzahl verdeckter Schichten (4):",
          v_schicht%
  ' pro verdeckter Schicht die Anzahl der
    Neuronen eingeben
  ' IF v_schicht%>0
  '   FOR i%=0 TO v_schicht%-1
  '     PRINT "Anz. Neuronen für verd. Schicht
              Nr.  ";i%+1;
  '     INPUT ": ",neuron_pro_schicht%(i%+1)
  '   NEXT i%
  ' ENDIF
  neuron_pro_schicht%(0)=eingangsneuronen%
  neuron_pro_schicht%(v_schicht%+
  1)=ausgangsneuronen%
  CLS
  PRINT AT(5,4);"Muster: ";
                 muster%
  PRINT AT(40 4);"Lernfaktor (eta): ";eta#
  PRINT AT(5,5);"Eingangsneuronen: ";
                 eingangsneuronen%
  PRINT AT(40,5);"Momentum (alpha): ";alpha#
  PRINT AT(5,6);"Ausgangsneuronen: ";
                 ausgangsneuronen%
  PRINT AT(5,7);"verdeckte Schichten: ";
                 v_schicht%
  PRINT AT(5,8);"Neuronen pro verd. Schicht: ";
  IF v_schicht%>0
    FOR i%=1 TO v_schicht%
      PRINT neuron_pro_schicht%(i%)''
    NEXT i%
  ENDIF
RETURN
'
PROCEDURE initialisieren
  LOCAL i%,j%
  durchgang%=0
  pruefen%=weiter%
  ' Ausgangsschwellen mit 1 belegen
  FOR i%=0 TO v_schicht%
    ausgang#(i%,neuron_pro_schicht%(i%))=1
  NEXT i%
  ' Gewichte mit Zufallswerten belegen
  FOR i%=0 TO v_schicht%
    FOR j%=0 TO (neuron_pro_schicht%(i%)+1)*
                neuron_pro_schicht%(i%+1)-1
      gewicht#(i%,j%)=RND(1)-0.5
      delta_gewicht#(i%,j%)=0
    NEXT j%
  NEXT i%
RETURN
'
PROCEDURE daten_einlesen
  LOCAL i%,j%
  FOR i%=0 TO muster%-1
    FOR j%=0 TO eingangsneuronen%-1
      READ eingang#(i%,j%)
    NEXT j%
    FOR j%=0 TO ausgangsneuronen%-1
      READ ziel#(i%,j%)
    NEXT j%
  NEXT i%
RETURN
'
PROCEDURE vorwaerts_propagieren(muster%)
  LOCAL j%,k%,l%,index%,netzeingang#
  ' den Ausgang der Schicht 0 bilden die
                                 Eingangswerte"
  FOR j%=0 TO eingangsneuronen%-1
    ausgang#(0,j%)=eingang#(muster%,j%)
  NEXT j%
  ' Netzeingang berechnen
  FOR j%=1 TO v_schicht%+1
    FOR k%=0 TO neuron_pro_schicht%(j%)-1
      netzeingang#=0
       FOR l%=0 TO neuron_pro_schicht%(j%-1)
         index%=(neuron_pro_schicht%(j%-1)+1)*k%+l%
          netzeingang#=netzeingang#+gewicht#(j%-1,
                 index%)*ausgang#(j%-1,l%)
      NEXT l%
      ' aktivieren
      ausgang#(j%,k%)=1/(1+EXP(-netzeingang#))
    NEXT k%
  NEXT j%
  FOR j%=0 TO neuron_pro_schicht%(v_schicht%+1)-1
    ausgabe#(muster%,j%)=ausgang#(v_schicht%+1,j%)
  NEXT j%
RETURN
'
PROCEDURE delta_ausgang_berechnen(muster%)
  LOCAL j%,aus#
  FOR j%=0 TO neuron_pro_schicht%(v_schicht%+1)-1
    delta#(v_schicht%+1,j%)=(ziel#(muster%,j%)-
        ausgang#(v_schicht%+1,j%))*(1-
        ausgang#(v_schicht%+1,j%))*
        ausgang#(v_schicht%+1,j%)
  NEXT j%
RETURN
'
PROCEDURE delta_vschicht_und_eingang_berechnen
  LOCAL j%,k%,l%
  FOR j%=v_schicht%+1 TO 1 STEP -1
    FOR k%=0 TO neuron_pro_schicht%(j%-1)
      delta#(j%-1,k%)=0
      FOR l%=0 TO neuron_pro_schicht%(j%)-1
        index%=(neuron_pro_schicht%(j%-1)+1)*l%+k%
        delta_gewicht#(j%-1,index%)=eta#*
            delta#(j%,l%)*ausgang#(j%-1,k%)+
             alpha#*delta_gewicht#(j%-1,index%)
        delta#(j%-1,k%)=delta#(j%-1,k%)+delta#(j%,
            l%)*gewicht#(j%-1,index%)
      NEXT l%
      delta#(j%-1,k%)=delta#(j%-1,k%)*(1-
          ausgang#(j%-1,k%))*ausgang#(j%-1,k%)
    NEXT k%
  NEXT j%
RETURN
'
PROCEDURE gewichte_aendern
  LOCAL j%,k%,l%
  FOR j%=1 TO v_schicht%+1
    FOR k%=0 TO neuron_pro_schicht%(j%)-1
      FOR l%=0 TO neuron pro schicht%(j%-1)
       index%=(neuron_pro_schicht%(j%-1)+1)*k%+l%
       gewicht#(j%-1,index%)=gewicht#(j%-1,
            index%)+delta_gewicht#(j%-1,index%)
      NEXT l%
    NEXT k%
  NEXT j%
RETURN
'
PROCEDURE fehler_berechnen(muster%)
  LOCAL j%
  fehler#(muster%)=0
  FOR j%=0 TO neuron_pro_schicht%(v_schicht%+1)-1
    fehler#(muster%)=fehler#(muster%)+
        ABS(ziel#(muster%,j%)-
        ausgang#(v_schicht%+1,j%))
  NEXT j%
  system_fehler#=system_fehler#+fehler#(muster%)^2
RETURN
'
```

→

```
PROCEDURE gewichte_und_schwellen_ausgeben
  CLS
  PRINT "Ausgabe der errechneten Gewichte und
        Schwellen"
  PRINT
  FOR i%=0 TO muster%-1
    vorwaerts_propagieren(i%)
  NEXT i%
  FOR i%=0 TO v_schicht%
    index%=0
    FOR j%=0 TO neuron_pro_schicht%(i%+1)-1
      PRINT "Gewichte zwischen Neuron ";j%;" der
            Schicht ";i%+1;
       PRINT " und Neuronen der Schicht ";i%
      FOR k%=0 TO neuron_pro_schicht%(i%)-1
        PRINT gewicht#(i%,index%)''
        INC index%
      NEXT k%
      PRINT
      PRINT "Schwelle des Neuron ";j%;" von
            Schicht ";i%+1
      PRINT gewicht#(i%,index%)
      INC index%
      PRINT "weiter = <Taste>"
      VOID INP(2)
    NEXT j%
  NEXT i%
  PRINT
  FOR i%=0 TO muster%-1
    FOR j%=0 TO ausgangsneuronen%-1
      form$="Vorgabe ### Ausgabe ##  = #.#####
            Ziel ## = #.#"
      PRINT USING form$,i%,j%,ausgabe#(i%,j%),j%,
            ziel#(i%,j%)
    NEXT j%
  NEXT i%
  PRINT
  PRINT "Durchgänge insgesamt ";durchgang%
  PRINT "Systemfehler   ";system_fehler#
  PRINT
  PRINT "weiter = <Taste>"
  VOID INP(2)
RETURN
'
FUNCTION abbruch
  LOCAL i%
  IF durchgang%>=max_durchgang%
    PRINT AT(5,24);"Abbruch mit Fehler -
            Durchgang > ";max_durchgang%
    VOID INP(2)
    RETURN (erfolglos%)
  ENDIF
  IF system_fehler#<=min_systemfehler#
    PRINT AT(5,24);"Abbruch erfolgreich -
            Systemfehler < ";min_systemfehler#
    VOID INP(2)
    RETURN (erfolgreich%)
  ENDIF
  PRINT AT(5,12);USING "Durchgang:     ####
            Maximum:  ####",durchgang%,
            max_durchgang%
  PRINT AT(5,14);USING "Abbruch bei: #.########",
            min_systemfehler#
  form$="Systemfehler: #.########  Abnahme: -
            #.########"
  PRINT AT(5,15);USING form$,system_fehler#,
            system_fehler#-alt#
  alt#=system_fehler#
  RETURN (weiter%)
ENDFUNC
'
PROCEDURE gelerntes_abfragen
  LOCAL j%
  CLS
  PRINT "Muster abfragen"
  PRINT "Werte eingeben"
  REPEAT
    FOR j%=0 TO eingangsneuronen%-1
      REPEAT
        PRINT "Eingabe ";j%;" (zw. 0-1): ";
        INPUT eingang#(0,j%)
      UNTIL (eingang#(0,j%)>=0) AND (eingang#(0,
            j%)<=1)
    NEXT j%
    PRINT
    vorwaerts_propagieren(0)
    FOR j%=0 TO ausgangsneuronen%-1
      PRINT USING "Ausgang ## = #.#####",j%,
            ausgang#(v_schicht%+1,j%)
    NEXT j%
    PRINT "Ende = <e>  weiter = <Taste>"
    REPEAT
      taste$=UPPER$(INKEY$)
    UNTIL taste$<>""
  UNTIL taste$="E"
RETURN
```



PD-JOURNAL-ABONNEMENT

Bücher / Software / ST Computer - Zeitschriften

erhalten Sie bei Ihrem Computer-Händler und im Buchhandel.

einsenden an:

Heim Verlag

Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon 06151-56057

Schweiz
DataTrade AG
Langstr. 94
CH - 8021 Zürich

Österreich
Haider
Computer + Peripherie
Granzer Str. 63
A-2700 Wiener Neustadt

Name/Vorname

Straße, Nr.

PLZ, Ort

Gewünschte Zahlungsweise bitte ankreuzen

☐ Bequem und bargeldlos durch Bankeinzug

Konto-Nr. BLZ

Institut Ort

☐ Ein Verrechnungsscheck über DM ________ liegt bei.

Diese Vereinbarung kann ich innerhalb von 8 Tagen beim Heim-Verlag, Heidelberger Landstr. 194, 6100 Darmstadt-Eberstadt widerrufen. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs. Ich bestätige die Kenntnisnahme des Widerrufsrechts durch meine 2. Unterschrift.

Datum Unterschrift

Ja, bitte senden Sie mir das Public Domain Journal

ab ________

für mindestens 1 Jahr (12 Hefte) zum ermäßigten Preis von jährlich DM 55,– frei Haus. (Ausland: Nur gegen Scheckvoreinsendung DM 75,– Normalpost). Der Bezugszeitraum verlängert sich nur dann um ein Jahr, wenn nicht 6 Wochen vor Ablauf des Abonnements gekündigt wird.

Datum/Unterschrift

AUSGESUCHTE und SORTIERTE Atari-ST Public-Domain-Software

Sehr geehrter Leser,
wir sind stolz darauf, Ihnen hier unser neues Public-Domain-Angebot vorstellen zu können! In dieser Form werden wir Ihnen in dieser Zeitschrift alle zwei Monate unser aktualisiertes PD-Angebot darbieten. Somit sind wir immer aktuell und erreichen hunderttausende interessierter Atari-ST-User.
Um lange Lieferzeiten zu vermeiden, liegen für Sie (dank unserer neuen Kopiermaschinen) über 6.000 bespielte PD-Disketten bei uns versandbereit!!
Jedes Paket, das unser Haus verläßt ist fehlerfrei (VERIFY), virenfrei (mehrfach getestet) und mit einem ausführlichen Aufkleber (Inhalt..) versehen!!

4 Diskettenseiten im Paket für ab DM 4.90
pro Paket incl. Disk

Alle von uns angebotenen PD-Pakete enthalten immer 4 komplette Diskettenseiten auf einer doppelseitigen 3,5"-Qualitätsdiskette (Made in Japan). Dies ist möglich durch ein revolutionäres Entkompaktierprogramm, welches sich auf jeder unserer Disketten befindet! Damit entkompaktieren Sie jeweils zwei Diskettenseiten in sage und schreibe in weniger als einer Minute (incl. entkompaktieren und kopieren)!! Die Bedienung ist kinderleicht und absolut sicher, da vollautomatisch. Schon tausende unserer Kunden sind begeistert: Denn außer des unschlagbaren Preises sorgt unser System auch für Ordnung und Platzersparnis in Ihrer PD-Sammlung!!!
Bitte achten Sie auch auf Tests verschiedener Fachzeitschriften (z.B. Joystick 7/89 und 8/89)!!
Noch Fragen?? Dann rufen Sie uns doch einfach mal an!!
Unter der Tel-Nr.: 07633/16994 sind wir täglich für Sie da!!!

Bestell-Nr.: 050030
>> TOP-PAKET DES MONATS (für alle ST geeignet) <<
Darauf hat die ATARI-WELT schon lange gewartet! Mit viel Mühe und Sachverstand haben wir Ihnen die 65 besten ACCESSORIES in einem prallvollen Paket zusammengestellt!!
(Accessories (*.ACC) = Programme, welche Sie von (fast) jedem Programm aufrufen können, ohne das jeweilige Programm abbrechen zu müssen.)

Hier ein paar Beispiele aus dem Inhalt unseres TOP-PAKET DES MONATS:
Spiele: Breakout, Puzzle, SGR, Mühle, ... /*/ Drucker: Druckeranpassungen, Hardcopys, Spooler, ... /*/ Verschiedenes: Rechner, Kopierprogramme, Uhren, Ramdisks, Fileselectboxen, Disk- und Maschinensprache-Monitore, Harddisk- und Diskettenhilfsprogramme, ...

Bestell-Nr.: 050001
>> TEST-Versionen (f. mind. 1 MB / S/W) <<
CALAMUS-DEMO = Calamus.... DAS beste DTP-Programm auf dem Atari-ST- erhalten Sie nun endlich bei uns als PD-Testversion!!
Bei diesem Programm handelt es sich um die NEUESTE VERSION des TOP-Programms CALAMUS. Jeder, der sich für Textverarbeitung, Grafik und DTP interessiert, muß dieses Programm mal erlebt haben. ALLE Funktionen dieses - in der ORIGINAL-Version weit über DM 700.— kosteten Programms - stehen Ihnen zur Verfügung. Die einzige Einschränkung: Sie können Ihre Werke nicht abspeichern!! Lassen Sie sich diesen Renner der Fa. DMC nicht entgehen!! /*/
Tristar = Demo-Version dieses tollen CAD- / Zeichenprogramms mit über 90 Funktionen!!
In der Testversionen sind drei (nicht unbedingt benötigte) Funktionen gesperrt. Ansonsten können Sie ganz normal arbeiten!!

Bestell-Nr.: 050002
>> ACTION-SPIELE-PAKET (S/W-MONITOR) <<
DGDB = Das große deutsche Ballerspiel. Schießen Sie, bis der Joystick raucht. /*/ 3D-Invaders = Ein Aktionspiel mit verblüffend schneller 3D-Vektorgrafik. /*/ Megaroid = Sehr gut gemachte ASTEROID-Variante /*/ Bounce = Es gibt Probleme auf der Erde. Nur Sie können sie beseitigen. /*/ Missile = Der 3. Weltkrieg bricht aus. Schützen Sie Ihre Städte vor einem Atomangriff. /*/ Invaders = Die Alien greifen die Erde an. Sie können sie retten. /*/ Cave Miner = Wenn sie Boulder Dash (TM) kennen, wissen Sie, was Sie zu tun haben. /*/ Pacman = Pillen fressen und Monster verjagen sind seine Hobbies. Helfen Sie ihm. /*/ Go Up = Sehr schöne Loadrunner(TM)-Variante. Sammeln Sie Schätze /*/ Maziacs = Finden Sie den Schatz im Labyrinth, achten Sie aber auf Ihre Feinde!!

Bestell-Nr.: 050003
>> ACTION-SPIELE-PAKET (S/W-MONITOR) <<
Opus 1 = Sie sind ein intergalaktischer Rennfahrer. Es gilt nun Nr. 1 zu werden. Rassantes, sehr realistisches 3D-Vektor-Rennspiel. /*/ Lander = Landen Sie Ihr Raumschiff auf dem Mond /*/ Bumerang = Haben Sie schon einmal mit einem Bumerang Break Out gespielt?? In diesem außergewöhnlichem Spiel können Sie den Bumerang sogar schneiden. /*/ Europa = Finstere Gestalten haben mehrere Bomben in Europa versteckt. Flitzen Sie in Ihren Gleiter und vernichten Sie alle Bomben.

Bestell-Nr.: 050004
>> SPIELE-PAKET (FARB-MONITOR) <<
DGDB = Die Farbversion des Ballerknallers. /*/ Durchbruch = DAS Break-Out-Spiel für Farbmonitore. /*/ Speedmaster = Wenn Sie Monster lieben, dann liegen Sie richtig mit diesem grafisch gelungenem Arcade-Game. Finden Sie alle Gegenstände und weichen Sie den Monstern geschickt aus. /*/ Megaroid = Farbversion des Spielhallenknüllern. Es gilt die Asteroiden sauber und schnell zu pulverisieren. /*/ Dragonhunt = Fangen Sie den Drachen, bevor er Sie fängt. Versuchen Sie den Ausgang des Labyrinthes zu finden. /*/ Bomber = Gut umgesetzte Cramble(TM)-Variante. Fliegen Sie so weit wie möglich in die Höhlen hinein, ohne abzustürzen. /*/ Taurus = Sie beherrschen eine Stadt. Mit dem Etat müssen Sie gut kalkulieren, um Ihre Stadt am Leben zu erhalten.

Bestell-Nr.: 050005
>> STRATEGIE-SPIELE-PAKET (SW-MONITOR) <<
Börse = Werden Sie zu einem Börsenspekulant und verdienen Sie Millionen. /*/ Corewars = Krieg der Kerne. Erstellen Sie selbst ein Kampfprogramm und lassen Sie es gegen andere Programme kämpfen und siegen. Den Kampf kann man direkt am Bildschirm miterleben. /*/ Fuffy = Wer kennt nicht Kniffel? Kniffeln Sie mit einem der besten Kniffelprogramme. /*/ Galerie = Eine hübsch gestaltete Patience-Variante. /*/ Memo = Der Computer gibt eine Tonfolge an, die sie nachspielen sollen. /*/ Ärger = "Mensch ärgere dich nicht". Eine sehr schöne Umsetzung des Spiels. /*/ Roulette = Wer gerne Roulette spielt, wird sich freuen. Gute Simulation mit toller Grafik /*/ Scribble = Ein gelungenes Scrabblespiel für 2 Spieler mit einfacher Bedienung. /*/ Solitaire = Hervorragendes 3D-Solitairespiel. Die Gestalt der Kugeln ist wählbar. /*/ Sprengmeister = Setzen Sie Ihre Steine vorsichtig aufeinander, sonst fliegen sie Ihnen um die Ohren. /*/ Happy 4 = Schön gestaltetes 4 Gewinnt-Spiel.

Bestell-Nr.: 050006 >> ADVENTURE-PAKET <<
Eb-City = Das au+ergewöhnliche Adventure mit einem sehr intelligenten Parser. Lassen Sie sich überraschen. /*/ Gilgalad = Ein rassantes AKTION-Adventure mit 224 Räumen. Vertreiben Sie die graue Eminenz. Doch Vorsicht, überall lauern Gefahren. /*/ Schlo+ = Ein SRRECHENDES Textadventure in deutscher Sprache mit ausgezeichneter Sprachausgabe. Sie müssen als Pink ein Dorf vor wilden Tieren schützen /*/ Sherlock = Wer hat den Gastgeber getötet. Sherlock, alias Sie, müssen diesen fast unlösbaren Fall anhand von Zeugenaussagen lösen. (Mit sehr schöner Grafik) /*/ Trash = Ein deutsches Textadventure der gehobenen Klasse. Sie müssen einen Hund retten, bevor ihn der letzte Mohikaner in seine Live-Show einbezieht. /*/ Vault = Deutsches Textadventure mit interessanter Story: Ein schottischer Schloßherr wurde an einem Baum erhängt aufgefunden. War es Mord? /*/ Larn = Ein Adventure mit EDITOR-Möglichkeiten. Dringen Sie in unbekannte Höhlen ein und machen Sie sich auf etwas gefaßt.

Bestell-Nr.: 050007
>> STRATEGIE-SPIELE-PAKET (SW-MONITOR) <<
Ballerburg = Schon einmal eine Burg zerstört? Nur wenn die Etatverwaltung stimmt, haben Sie eine Chance über Ihren Gegner zu siegen /*/
Dallas = Das flüssige Gold müssen Sie sich in diesem Spiel erst erwirtschaften. Ihre gegnerischen Spieler können Ihnen Schwierigkeiten machen. (f. 2 bis 8 Spieler) /*/ Hotelier = Als Hotelier hat man's schwer. Die Gäste wollen einfach nicht kommen. Nur durch Ausbau Ihres Hotels können Sie Ihre Kunden wiedergewinnen. Es gilt auch die Konkurenz auszuschalten. /*/ Monopoly = Die Schlo+allee ist weit. Zuerst sollten Sie Grundstücke kaufen, Mieten von Ihren Mitspielern einziehen und Häuser bauen. /*/ Fugger = Sie als der Nachkomme des Jakob Fugger müssen das heruntergekommene Geschäft wieder flott machen. Eine schwere Aufgabe. /*/ Metropol = Versuchen Sie 30 Jahre lang ein Volk zu regieren ohne abgesetzt zu werden /*/ Football Manager = Managen Sie einen Fu+ballclub von der 3. Liga ab. Eintrittspreise erhöhen, Spieler einkaufen, usw.

Bestell-Nr.: 050009
>> UTILITY-PAKET COMPUTER / DISK / FESTPL. <<
Filebox = verbesserte Fileselectbox. 16 Laufwerksymbole direkt anwählbar, Anzeige der Prg.-Länge, -Zeit, -Datum /*/ WT-HD = Dateisuchprogramm für Festplatte, geschwindigkeitsoptimiert, Suche mit Wildcards /*/ SAKROTAN 4.1 = Dieser Viruskiller (der BESTE unter PD) erkennt und löscht Viren, legt Vergleichsdateien an, erstellt Virusschutzprg. /*/ Filecode = Verschlüsselt Dateien /*/ ASCII ED = Komfortable Umbelegung der Tastatur /*/ Bootwähler = Bootprg zum Auswählen aus mehr als 6 ACC /*/ Quickmouse = Beschleunigt Mouse /*/ Temelmon = Guter Monitor / Debugger, der durch Alternat/Help aufrufbar ist /*/ Bitte ein Bit = Das beste uns bekannte PD-Kopierprogramm mit automatische Formaterkennung und Umformatierung ,hoher Kopiergeschwindigkeit, usw., usw., usw. /*/ Format = Multitasking-Formatierprogramm (formatiert im Hintergrund) /*/ MC87_1_9 = Formatier-, Kopier-, Harddiskback-Programm /*/ Monomon = Monitor für Floppy, Festplatte und Ramdisk, mit vielen Funktionen (z.B. = direktes Anwählen der FAT / Direktory / Bootsektor....) /*/ Totaldel = Vollständiges Löschen von Dateien (physikalisch) /*/ Make1MB / Make512KB = Computer auf 1MB/512KB einstellen /*/ Speed = Floppy-Beschleuniger

Bestell-Nr.: 050010 >> UTILITY-PAKET DRUCKER <<
Fontprinter = Erzeugung von 9-Nadel Downloadfonts mit Editor in Draft + NLQ, mit automatischer Installation /*/ Hardcopy = Bildschirmausdruck in variabler Grö+e (DIN A8 bis A0), für 24 oder 9 (8) Nadler, in rechten Winkeln drehbar/*/ Posterprint = Bildschirm in Posterformat ausdrucken /*/ Big Banner = Erzeugung von meterlangen Bannern in versch. Schriftarten und -Größen /*/ NECP2002-INST / NL10-INST = Einstellprg für sämtliche Steuercodes per ACC /*/ Sticker = Disketikettendrucker, übernimmt PRG-Namen in Etiketten. Bilder von versch. Gebieten vorhanden. Mit integriertem Zeichenprogramm /*/ Druckertreiber für 1st Word+(TM) = Citizen, NL10, Gabigraph, LC10

Bestell-Nr.: 050110
>> UTILITY-PAKET DRUCKER 2 <<
Kuvert = professioneler Schnelldruck von Adressen und Briefumschlägen /*/ Power = neuer Zeichensatz für den ST (die Schrift bleibt nach verlassen des Programmes erhalten /*/ PrintDir = schneller Ausdruck von Directories /*/ ETI Master = Etikettendrucker , kann auch als ACC geladen werden /*/ Geheim = ver- und entschlüsselt Daten /*/ Überwei = druckt Überweisungsformulare /*/ Tempdemo = Demoversion des SUPER Texteditors " TEMPUS " /*/ CED = bedrucken und herstellen von Kassettenhüllen /*/ ZSEditor = Zeichensatzeditor für 24 Nadeldrucker , druckt aus jedem Programm mit dem neuen Zeichensatz /*/ Adress = ein sehr einfach zu bedienendes Adressverwaltungsprogramm mit Sortierfuktion /*/ Paste = druckt Adressaufkleber im Format 1,5 cm * 2,5 cm !! Verschidene Schriften . Läuft auf allen Druckern. (inklusiev Druckeranpassung) /*/ Label = Diskettendrucker übernimmt Bilder per Mausklick /*/ Datei = flexibele Dateienverwaltung , freie Maskenerstellung per Dialog , Daten im und export und Steuerzeichen anpassung /*/ Filecode = schützt Ihre Dateien vor unbefugtem Gebrauch /*/ XMODEM = das absolute muss für jeden Modembesitzer

Bestell-Nr.: 050011 >> BUSINESS-PAKET <<
Verein = Mitgliederverwaltung. Eingabemögl. von Adresse, Geburtsdatum... /*/ Meda = Dateiverwaltung mit freier Maskenerstellung, Anbindung von Serienbriefen durch 1st-Mail(TM) /*/ Probase = Prof. Datenbank (beste PD-). Freie Maskenerstellung mit unterschiedlichen Feldtypen (Zahl, String, ..), Passwortschutz /*/ PD-Text = Bestes PD-Textverarbeitungs-Programm mit allen gängigen Funktionen, wie Textattribute, Wortumbruch, Formatieren in Block und Flattersatz, Blockoperationen, ... /*/ Scheck = Druckprg für Schecks und Überweisungsaufträge /*/ Artus = Verwaltung von Artikeln aus Zeitungen und Zeitschriften. Auswahl nach Stichpunkten, Namen, Monat u.a. /*/ Datei = flexible Datenverwaltung. 19 Felder mit je 99 Zeichen /*/ Zeitmanager = Terminverwaltung mit vielen Funktionen. Suchen nach Stichworten, Daten... Notizen ausdruckbar oder zu einer Ideensammlung zusammenfaßbar

Bestell-Nr.: 050012 >> BUSINESS-PAKET <<
Adr_2 = Adressverwaltung. Erstellen eines Telefonbuches mit Geburtstagsliste. Serienbrief und Sotieren nach Kriterien ist möglich /*/ Adressen = Adressverwaltung, Etikettdruck und Texteditor. Serienbrief möglich. Mit komfortabler Sortierroutine /*/ K_Kasten = Verwaltungsprogramm für Adressen, Briefmarken, Musikcassetten, Videos... /*/ Selector = Dateiverwaltung mit vielseitiger Sortierroutine /*/ Adress = Private Adressverwaltung. /*/ Finanzmathe = Berechnung von Zinsen, Laufzeiten, Tilgungsplänen, usw. /*/ Aktie = komfortable Aktienverwaltung /*/ Anlagen = Verwaltung von Anleihen, Aktien, Investmentanteilen, usw. /*/ Depot = Aktiendepot für Finanzgenies /*/ Fibu = Gutes Finanzbuchhaltungsprogramm /*/ Versand = Lohnbuchhaltung und Versanddruck mit vielen Extras /*/ ST-Calc = Gutes Tabellenkalkulationsprogramm mit 2600 Zellen und vielen Funktionen

Bestell-Nr.: 050013 >> HEIMANWENDUNS-PAKET <<
1stStep:Kalender und Biorythmus /*/ Datebook = Erinnert bei Systemstart an Termine, Geburtstage.. /*/ ETAT / Hausvg = Planung und Kontrolle des Haushalts. Buchung, Monats-/Jahresplanung, Kostenverwaltung, Jahresabrechnung, usw. /*/ Genius = Reaktionstest und Gedächtnisschulung mit Auswertung der Ergebnisse /*/ IQ-Test = Berechnet nach grafischen und mathematischen Tests Ihren IQ-Faktor /*/Urlaub = Berechnung der schönsten Tage im Jahr /*/ Waschi = Hilfsprg für Junggesellen. Hilft bei der Wäsche und bei der Auswahl der Waschprogramme /*/ Alktest = Berechnet unter Berücksichtigung der Körperma+e den theoretischen Blutalkoholwert, zeigt Abnahmekurve, informiert über Gefahren

Bestell-Nr.: 050033 >> DFÜ-PAKET <<
Das läßt die Herzen aller DFÜ-Freundehöher schlagen!! Speziell für die Daten-Fern-Übertragung (DFÜ) haben wir alles Gute zusammengesucht und in ein Paket gepackt: Jam-Term = Gutes Terminalprogramm der Infobox in Berlin /*/ Moses = Leicht zu bedienendes Terminalprogramm /*/ MCS = sehr interessantes MAILBOX-CONSTRUKTION-SET /*/ Kermit = Ein sehr bekanntes Terminal-Programm unter GEM /*/ S-Term-D = Demoversion des kommerziellen Programms /*/ Contact / STERM = Sehr kurze DFÜ-Programme /*/ Hack = Programm für den Hacker /*/ CON = Schnelles Terminalprogramm * St-Phone = Telefonkostenberechnung * Termin = Sehr konfortables Terminal-Programm /*/ Term = ...mit X-Modem /*/ TGS = für Hacker * Porno-DFÜ = Terminal-Programm mit Erotik-Einlage /*/ Uniterm = Ein wirklich sehr konfortables Terminal-Programm mit einer Langen latte nützlicher Funktionen /*/ BTX-Desk = Hilfsprogramm für BTX-Beistelldecoder mit vielen automatischen Funktionen

Bestell-Nr.: 050014
>> LERN- und SCHULPROGRAMME <<
Funccalc = PRG zur numerischen Mathematik. Eingabe und Berechnung von Variablen und Funktionstermen. Hohe Geschw. durch Maschinensprache. Simpson-Integralberechnung /*/ Plotter = Komfortabler Kurvenplotter. Bis zu 5 überlagerbare Kurven mit je bis zu 20 Variablen, Ausschnittsvergrö+erung, Raster, Texteinfügung, Zeichenfunktionen... /*/ Schreibmaschine = Sehr guter Schreibmaschinenkurs zum Erlernen des 10-Finger-Systems. Eigener Kurs editierbar /*/ Quicklern = Vokabelprg der Spitzenklasse mit sehr guter Benutzerführung. Erstellen eigener Lernbibliotheken /*/ Artus / Kopfrechnen = Rechentrainer für untere Schulklassen, führt mit netten Auflockerungen das Thema Mathematik heran /*/ Rechnen = Rechnen für Bürokaufleute. Umrechnen von Ma+einheiten, Prozentrechnen, usw. /*/ Stundenplan = Stundenplandesigner für Schüler und Lehrer /*/ World = Fragt alle Länder und Hauptstädte der Welt ab. Mit Landkarten von BRD / USA / Amerika / Europa / Asien / Afrika / Ozeanien /*/ Körper88 = Berechnungen von Würfel / Prisma / Zylinder / Pyramide / Kegel / Kugel / Raute / Dreieck / Trapez / Kreis / usw....

Bestell-Nr.: 050034
>> GRAFIK-ANWENDUNGEN (SW-MONITOR) <<
SIM-Draw = Ausgezeichnetes, sehr umfangreiches Malprogramm aus dem Ausland /*/ Convert = Bit-Image-Converter /*/ BW-Grafik = Gutes Malprogramm mit tollen Möglichkeiten /*/ Paintlux = Extrem luxuriöses Malprogramm mit einer Latte von Extras und einer guten Dokumentation /*/ AIM = Komplexes Bildverarbeitungs-Paket aus Holland. /*/ Little = Malprogramm mit NEC-P6-Treiber und fantastischen Funktionen PictureST = Malprogramm mit 4 Ebeben

Bestell-Nr.: 050035
>> GRAFIK-ANWENDUNGEN (FARBE) <<
MINI-CAD = Erlaubt das erstellen 3-Dimensionaler Farbgrafiken * Neochrome = Eines der ersten - und doch immernoch der besten Malprogramme für Farb-Monitorbesitzer /*/Masterp = Konvertiert jede Auflösung in eine beliebige Andere /*/ Tenstar+ = Tolles Zeichenprogramm, das selbst profesionelle Programme um längen schlägt. Mit diesem Demo können Sie uneingeschränkt zeichnen, nur nicht abspeichern. Mehr Funktionen als jedes andere Zeichenprogramm, incl. 3-D!!! /*/ Van Gogh = Sehr shönes Mal- und Animationsprogramm mit dem Sie bis zu 64 Sprites gleichzeitig darstellen können.

Bestell-Nr.: 050036
>> GRAFIK-ANWENDUNGEN (SW-MONITOR) <<
Megapaint = Demo eines Zeichenprogramms der Spitzenklasse, das Ihren ST in einen prof. Grafikterminal umwandelt. Dieses Programm bietet ungeahnte Möglichkeiten. /*/ PPM = Malprogramm mit interessanten Möglichkeiten /*/ MF = Gestalten Sie mit diesem wirklich eindrucksvollen Programm Ihren eigenen Trickfilm. Sie zeichnen einfach die Anfangsposition und Endposition einer Figur. Der Computer berechnet die Zwischenschritte selbst!! /*/ AR-CAD = Ein Programm direkt aus Hollabnd, das einiges zu bieten hat.

Bestell-Nr.: 050017 >> MUSIK- UND MIDI-PAKET <<
MIDI-TEIL = Midi macht aus Ihrem ST eine Rhythmusmaschine. Bis zu 5 Schlagzeugklänge können aus einem D/A-Wandler erklingen. Natürlich gibt es viele Möglichkeiten den Sound zu editieren und zu bearbeiten: Einstellung der Zählweise, 32 Loops möglich, Punch in/out, Bandzählwerk, Metronom, Notengrafik, Editor für Tonhöhen-/Längen-/Dynamik-Daten, spezieller Drumeditor und jede Menge Kopierbefehle. Auch können beliebige Mididaten empfangen und auf Diskette gespeichert werden. Ein weiteres Midi-Programm nimmt in Realtime Ihr Musikstück auf und speichert es auf Diskette. Besonders lobenswert ist, daß die gespeicherten Daten als Notenschrift ausgedruckt werden können. /*/
MUSIK-TEIL = Deluxe Piano ermöglicht das einfache Erstellen von Sequenzen. Es setht eine 2-oktavige Tastatur zur Verfügung. Der Akkord oder die Sequenz läßt sich bequem und vom Benutzer über die Soundparameter (Envelope, Waveform, Mixer) frei eingeben, speichern und laden. Mit Musik können Sie ganze Lieder im Xbios (32)-Format (also in jede Programmiersprache einbaubar) komfortabel, einfach und schnell komponieren. Die Moteneingabe erfolgt einfach durch Maus. Vorzeichen, Triolen, Länge der Pausen, Abspieltempo und Lautstärke sind natürlich frei einstellbar; umfangreiche Kopierroutinen sind ebenfalls implementiert. /*/ X32_Conv = Wenn Sie in Musik ein Musikstück erstellt haben, so können Sie es mit diesem Programm in Xbios(32)-Format umwandeln lassen. Soundcomputer erleichtert das Erstellen von Soundeffekten für Ihre Programme. Es ist möglich, alle Register des Soundchips anzusprechen (Tonhöhe, Lautstärke, Rauschen, Periodendauer...). Mehrere fertige Soundeffekte sind bereits direkt abrufbar. Für GFA-Basic-Besitzer werden die erforderlichen Programmzeilen angezeigt.

EROTIK-PAKETE: Diese Pakete werden nur an Kunden ab 18 Jahren ausgeliefert!! Jede Diskette enthält freizügige Grafiken von VIELEN Pin-Up-Girls in hoher detailgetreuer Aufläsung und tollen Farben. Jede Diskette enthält verschiedene Bilder, es gibt KEINE ÜBERSCHNEIDUNGEN. Ein wirklich toller Gag für Ihre nächste Party

Bestell-Nr.: 053006 >> EROTIK-PAKET <<
(bei Bestellung bitte Geburtsdatum angeben!!)

Bestell-Nr.: 053007 >> EROTIK-PAKET <<
(bei Bestellung bitte Geburtsdatum angeben!!)

Bestell-Nr.: 053008 >> EROTIK-PAKET <<
(bei Bestellung bitte Geburtsdatum angeben!!)

PROGRAMMIERSPRACHEN: Da flippt doch der kuhlste Programmierer aus... denn jetzt bieten wir allen Atari-ST-Freaks unsere neuen Programmiersprachen-Pakete an!! Jedes Paket wurde mit den neuesten PD-Versionen ausgesuchter Programmier-Sprachen gefüllt. Jede Programmiersprache wird incl. meist deutscher und ausführlicher Anleitung auf Diskette geliefert. Beispiel- und Hilfsprogramme sind natürlich auch dabei!! Da muß jeder zugreifen!!!

Bestell-Nr.: 050018
>> PROGRAMMIERSPRACHEN-PAKET <<
Modula 2 = Eine der interessantesten und NEUESTE Programmiersprachen auf dem Atari ST. >>> Wir liefern die NEUESTE VERSION <<< (komplett auf 2 Diskseiten) /*/ für obiges Programm /*/ TexteditorAssembler = 2-Pa+-Assembler, Disassembler, Debugger, Editor

Bestell-Nr.: 050019
>> PROGRAMMIERSPRACHEN-PAKET <<
Volks-Forth = Komplettes Forth-System (neueste Version auf 3 Diskseiten) /*/ LOGO = Logo - jetzt auch auf dem Atari ST (komplett auf 1 Diskseite)

Bestell-Nr.: 050020
>> PROGRAMMIERSPRACHEN-PAKET <<
"C" = Die beliebte Programmiersprache C jetzt endlich auch als Public-Domain im Komplettpaket /*/ Prolog = Diese Sprache der neuesten Generation /*/ XLisp = Neueste Version dieser Programmiersprache der KÜNSTLICHE INTELLIGENZ (KI) /*/ Expert = Dieses Expertensystem erweitert sein Wissen im Dialog mit dem Anwender. Damit wird ermöglicht, Wissensbanken gewisser Themensgebieten anzulegen.

Bestell-Nr.: 050032
>> PROGRAMMIERSPRACHEN-PAKET <<
Little Smalltalk = Eine selbstständige Programmiersprache der Universität Arizona. Diese wirklich hochinteressante Programmiersprache sollte sich wirklich jeder Atari-Programmierer mal ansehen. /*/ Pascal-Comp = Dieses Programm ist eine Mischung zwischen den beiden beliebtesten Programmiersprachen PASCAL und "C". Diesen wirklich außergewohnlichen Compiler sollte sich wirklich jeder Atari-Besitzer mal ansehen /*/ Icon = Diese aus den USA stammende - eigenständige - Programmiersprache versteht sich als Nachfolger von "SNOBUL 4"

Bestell-Nr.: 050021
>> PROGRAMMIERUTILLITIE-PAKET <<
BasicAESLIB = Komplette AES-Bibliotek für GFA-Basic mit Linker /*/ Basic Conv = Wandelt ST-Basic-Prgs in GFA-Basic um /*/ Spriteed/Iconedit = Komfortabler Spriteeditor für GFA-Basic, Editorfunktionen wie Scrollen, Spiegeln, Drehen, usw. /*/ Seco = Verwaltet Bildausschnitte (mit GFA-Basicbefehlen GET/PUT) und fügt sie dann zusammen /*/ GFA-Shell = ein Shell für alle GFA-Programmierer. CLI und Spriteditor eingebaut /*/ BasicXref = Crossreferenzprg für GFA-Basic. Basic-Prgs. untersuchen, Variablenliste erstellen und drucken /*/ C Library = einige umfangreiche Bibliotheken über Line_A, Floating, Fileorganisation, verbesserte Fileselectroutine... /*/ C Xref = Crossreferenzprg für C-Programmierer, alle Variablen- und Funktionsnamen werden angezeigt. Dadurch vereinfachte Anpassung systemfremder Programme /*/ Command = Commandointerpreter: Dateien packen, entpacken, verschlüsseln, Find, Passwortabfrage, History-Funktionen, Batchprozessing... /*/ Gulam = Shell für Modula-Programmierer, DOS-ähnliche Oberfläche, 61 feste Befehle, Funktionsaufrufe selbst editierbar /*/ Flesh = erzeugt ein Shell, mit dem Sie jedes Programm per Menüpunkt anwählen können /*/ Pascal Complex / Mathelib = umfangreiche Mathelibrary für Modula-2. (Source dokumentiert)

ARTWARE: Geschlagene 32 prallvolle Disketten (1-seitig / 16 Disks DS) feinster Grafiken umfasst momentan unser Artware-Programm. Die ca. 9000 ausgesuchten Grafiken sind im Degas-Format abgelegt. Sie können mit dem mitgelieferten Konvertprogramm in alle gängigen Formate umgewandelt werden (für SIGNUM, STAD, und alle DTP-Progr.). Greifen Sie auf diese umfangreiche und qualitativ hochwertige Grafik-Serie zurück... Sie werden sie 100-fach nutzen!!

Bestell-Nr.052002
>> SIGNUM(tm) - ZEICHENSATZ-PAKET <<
ARTDECO * BABYTEET * BURLING * DATA70 * FINO * FOX / FUCHS * PIONEER * ROCKWELL * SÜTT9 * SCHABLON * SCHWAB * WORMNORM * ABC * ALPINE * ALTDEUTI * ALTGRIE * AMBER * AMELIA * ANNA * ANTIK4+5 * ANTIK NP * ANTIMKR * ANTI CAP * AVANTGAR * CALIGRAF * CHANTAL * CIRCULUS * COMPUTER * GOTHIK * HAND * HAND2 * HITECH26 * KÜNSTLER * MODERNE * NOUTLINE * PEKING * UNIVERSE

Bestell-Nr.: 052003
>> SIGNUM(tm) - ZEICHENSATZ-PAKET <<
BÖCKLIN * BAUHAUS * BLOCK GR * BUBBLE * CASLON * CHEMIE * CHEMIE1 * CHEMIE 2 * CHEMIE * COMOCHAR * COMPACTA * COMPUTER * COMPU FT * COMPU LT * COURFE * COURIELT * COURLT * DATA70 * DECO * DESIGN * EGYPTIEN * ARTDECO * BABYTEET * FRAKTUR1 * FRACTUR2 * FARNZGRF * FUT * FUTURA * FUTURBIG * GOTHIK * GRAFIK * GRAPH1 1 * GRAPH3 * GRAPH38 * GRAPH4 * GROTBIG F * GROT CAP * HARRINGTON * HEBRAE * HEBRFETT * HEBRHELL * JAPAN

Bestell-Nr.: 052004
>> SIGNUM(tm) - ZEICHENSATZ-PAKET <<
KARIN * KINDER * KYRILLGR * LABIG * LANORM * LCD * MARCUS * MATHEM2 * MELODIE1 * MODERN * MUSIC * NEWYORK * NORMAND2 * NORMALI * PCOURFE * PCOURLT * PICTURE * PLOP * PRIS II * PUMPBIG * PUMPBOLD * PUMPLIGH * PUMPMED * QUADRATO * PIONEER * ROCO * ROKWELL * RUSS 2.D * SÜTT * SÜTT 24 * SCOLA * SCRIBA * SCRIBERE * SCRITTO2 * SERIFLT * SERIFMKR * SMALL * STRIKE * STRUKT2 * SYMBOL * UNIROMBI * UNIROMME * UNIVER11 * UNIVER14 * VABIG * VEKTOREN * VIZA * WERNER

Bestell-Nr.: 052005
>> SIGNUM(tm) - ZEICHENSATZ-PAKET <<
ANTROBL7 * PHONETIK * BASKVIL * CARELIT1 * CAPRICA2 * FRAKTURV * IGROTFERE * SYMBOLS * PRIMUS * KELTIC * CHRISMKR * CHRIST 1B * ANTIQUA * BROADWAY * COPPERB * FALSCH * GARAMONT * JUGEND2 * LINEAR * LINIEHOR * OXFORD * PEIGNOT * PLAKAT * SUPASTAR * TIMES * YUPPIE * ZAPF

Bestell-Nr.: 052006
>> ARRTWARE-BILDER-PAKET <<
MASCHINEN * ROUTENSILIEN * EINKAUF-VERKAUF * WERKZEUG * MÖBEL * GELD * SCHULE * SCHMUCK * WIRTSCHAFT * ROARTIKEL * ORDEN * TELEFON * LEBENSMITTEL * VERKEHRSZEICHEN * UHREN * TYPO * HÄNDE * MEDIZIN * STÄDTE * MENSCHEN * UMGESETZTE SPRICHWÖRTER * COMICFIGUREN * SPORT

Bestell-Nr.: 052007
>> ARRTWARE-BILDER-PAKET <<
FAHRZEUGE * BERUF / HANDWERKSZEUG * TIESAMMLUNG

Bestell-Nr.: 052008
>> ARRTWARE-BILDER-PAKET <<
TIERE * LANDKARTEN * UMRI+ZEICHNUNGEN * TYPO * SYMBOL * RAHMEN

Bestell-Nr.: 052009
>> ARRTWARE-BILDER-PAKET <<
FONTS * RAHMEN * SYMBOLE * MUSIKINSTRUMENTE * STERNZEICHEN * SPORT * MENSCHEN * FIGUREN * COMIC * HINWEISSCHILDER * BÄUME * INDIANER * NOTIZEN * FESTE * RELIGION * TIERE * ESSEN * VERKEHR * STÄDTE

Bestell-Nr.: 052010
>> ARRTWARE-BILDER-PAKET <<
KONVERT. APPLE-MAC.-ARTWORKS HITECH * MUSIKINSTRUMENTE * PERSÖNLICHE UTENSILIEN * WOMAN-SERIE * SYMBOLE * MAX-HEAD * CELTIC * TIERE * MYTHISCHE WESEN

Bestell-Nr.: 052011
>> ARRTWARE-BILDER-PAKET <<
KONVERT. APPLE-MAC.-ARTWORKS Diskette Nr. ATARI * S-CLIP1 (ca. 150 icongroße Bilder) * ALTE WARENZEICHEN AUS USA * BLUMEN * ORNAMENTE * ESSEN * AUGEN UND GESICHTSAUSDRÜCKE IN ALLEN VARIATIONEN * ROBO-CLIP * CARTOON

Bestell-Nr.: 052012
>> ARRTWARE-BILDER-PAKET <<
KONVERT. APPLE-MAC. * SPORTSACHEN1 * FREIZEIT * SPIEL * URLAUB * SEHENSWÜRDIGKEITEN * WEIHNACHTEN * WEIHNACHTSMANN * KERZEN * WINTERLANDSCHAFT * HALOWEENSERIE 1-3 * FESTSPEISEN * FRÜHSTÜCK * SILVESTER * AMOR-LIB * AMERIKA

Bestell-Nr.: 052013 >>ARRTWARE-BILDER-PAKET <<
WEIHNACHTEN * GRATULATION (für Hochzeit, Nachwuchs, Geburtstag ...) * REISEN * MAFALDA-SERIE * BERUF * MANNI-SERIE 1-9 * MISTER 1-3 * POLITIKERKÖPFE

Bestell-Nr.: 052014 >>ARRTWARE-BILDER-PAKET<<
TIERE 1-5 * ANZEIGEN * HASE 1-3 * RÄUBER 1-3 * ZAUBEREI 1,2 * SPORT-PICTOGRAMME * UNIFORM-PICTOGRAMME * ANZEIGEGESTALTUNG * SCHLAUER FUCHS * 10 DM-SCHEIN * URKUNDEN * JAHRESZEITEN-PICTOGRAMME * LAMPENGEIST-ANZEIGEFORM 1,2 * GESICHTER 1,2 * COMIC-GERÄUSCHE

Bestell-Nr.: 05XXXX >> ÜBERRASCHUNGS-PAKET <<
Beinhaltet ein Paket nach UNSERER WAHL /*/ Für Leute, die sich nicht entscheiden können, machen wir durch unseren absoluten KNULLERPREIS die Entscheidung leichter: NUR DM 4.90 bezahlen Sie pro Paket!! (= DM 1.22 pro Diskseite) /*/ Wenn Sie mehrere ÜBERRASCHUNGSPAKETE bestellen, garantieren wir Ihnen, daß kein gleiches Paket vorkommt. (Abgabe: maximal 10 PD-Pakete pro Besteller!! / Zählt NICHT zur Preisstaffel!!)
ZEITSCHRIFTEN-PD-Pakete:
Außerdem liefern wir die komplette PD-Serie einer gro+en deutschen ATARI-ST-COMPUTER-Zeitschrift!! (PD-Neuerscheinungen ca. 7 Tage nach Erscheinen lieferbar) Unterlagen anfordern!!
ZEITSCHRIFTEN-PD-GESAMTANGEBOT:
Die komplette Serie unserer Zeitschriften-PD-Pakete (Disk Nr. 1 bis Disk Nr. 224) auf 56 Disketten (Paketen) können Sie bei uns zum SUPER-KNULLER-PREIS von NUR DM 302.40 (ergiebt DM 1.35 pro Disk-Nr.) bestellen!! Besonders interessant für Clubs, PD-Neulinge und EDV-Handlungen!!!
PD-ABO: Bitte achten Sie auf unseren sensationellen ABO-Service im PD-JOURNAL vom HEIM-Verlag
Preise & Lieferbedingungen:
Jedes Paket mit je 4 Disk-Seiten pro Paket DM 8.00
ab 10 Paketen pro Paket DM 7.50
ab 20 Paketen pro Paket DM 7.00
ab 30 Paketen pro Paket DM 6.50
Bestellungen unter einem Mindestbestellwert von DM 30.— können wir aus Rentabilitätsgründen leider nicht bearbeiten!! (IRRTUM VORBEHALTEN /*/ Mit Erscheinen dieser Anzeige verlieren alle älteren Angebote Ihre Gültigkeit!!!!!)
Porto und Verpackung:
Bei Lieferung per Nachnahme : DM 6.00
Bei Vorauszahlung (V-Scheck

FsKs D. Ludwig • Mozartweg 8 • D-7812 Bad Krozingen • Telefon: 07633/16994

Programmierte Logik

Über GALs und ihre Programmierung

Wenn man eine logische Schaltung aufbauen und entwerfen will, kann man auf die altbewährte TTL-Logik zurückgreifen, die mittlerweile eine umfangreiche Standardbibliothek an Bausteinen mit vielen Grundfunktionen zur Verfügung stellt. Hat man sich für die benötigten ICs entschieden, muß man die einzelnen Logikgatter untereinander verbinden. Das führt zu einem mehr oder weniger großen Platzbedarf auf der Platine und kann, in der Entwicklungsphase, zu manch unangenehmen Stunden der Fehlersuche führen (wer Logikschaltungen bereits selber entworfen und aufgebaut hat, kann davon ein Liedchen singen). Von den Schwierigkeiten, die man bekommt, wenn sich die Logik später leicht ändern oder eine Modifikation notwendig werden sollte, rede ich lieber gar nicht erst.

All diese Schwierigkeiten und Probleme hat man nicht, wenn man sich dafür entscheidet, ein GAL zu benutzen. Die Vorteile liegen auf der Hand. Mit einem GAL kann man mehrere Standardbausteine einsparen (ersetzt bis etwa 6 TTL-/CMOS-ICs). Damit entfällt dann auch die Entwicklungszeit für die Verdrahtung der Gatter untereinander. Außerdem kann die Schaltung kompakter aufgebaut werden. Desweiteren sind GALs lösch- und neuprogrammierbar (es wird eine Datensicherheit von etwa 20 Jahren bis zur 100. Löschung garantiert).

Was sind eigentlich GALs?

GAL ist eine Abkürzung und steht für *G*eneric *A*rray *L*ogic. Sie werden benutzt, um Logikgleichungen hardwareseitig zu realisieren, zum Beispiel als Adreßdecoder. Nehmen wir an, ein Chip in einem Computer wird unter einer bestimmten Adresse angesprochen. Dann würde man an die Eingänge des GALs alle benötigten Adreßleitungen anschließen und die interne Logikmatrix so programmieren, daß der benötigte Ausgang (als Chip-Select-Signal, d.h. als Leitung, die nur dann *high* wird, wenn der Baustein angesprochen werden soll) dann aktiv wird, wenn die Adresse des Bausteins am GAL anliegt. Und wie programmiert man so ein GAL? Nun, als erstes muß man die Logikgleichungen aufstellen, d.h. man muß festlegen, bei welcher Adresse der

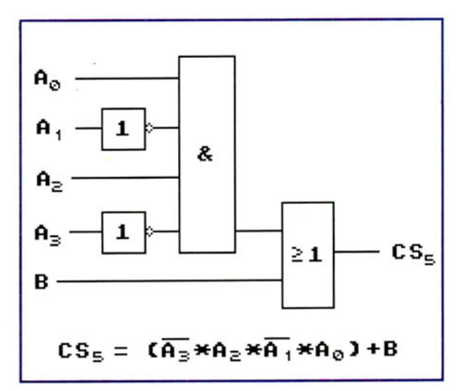


Bild 1: Unser Beispiel in TTL-Logik

Baustein angesprochen werden soll (um beim obigen Beispiel zu bleiben). Wir wollen der Einfachheit halber einmal annehmen, daß in unserem Beispiel die Bausteine extra Adressen zwischen 0 und 15 haben, d.h. wir haben vier Adreßleitungen (A_0 (Low-Bit) bis A_4 (High-Bit)). Das IC liege bei Adresse fünf, was binär 0101 entspricht. Daraus erhalten wir unsere Logikgleichung zur Ansteuerung des Bausteins, wenn wir das Chip-Select-Signal mit CS5 bezeichnen, zu (*: UND; /: NICHT):

$$CS_5 = /A_3{*}A_2{*}/A_1{*}A_0$$

Das IC soll aber auch über einen anderen Baustein angesprochen werden können, d.h. auch dann, wenn der andere Baustein ein Signal (B) sendet, muß das Chip-Select-Signal gesetzt werden. Dies erweitert unsere Gleichung zu (+ = ODER):

$$CS_5 = (/A_3{*}A_2{*}/A_1{*}A_0)+B$$

Jetzt haben wir unsere fertige Gleichung und können diese sowohl mit Standardbausteinen (Bild 1), als auch mit einem GAL realisieren. Zur Realisierung mit einem GAL sehen wir uns diese erst einmal näher an.

So sehen sie aus!

In Bild 2 und Bild 3 ist die Logik-Matrix der beiden GAL-Bausteine *16v8* und *20v8* wiedergegeben. (Ich werde bei der Beschreibung die Daten für den 16v8 nennen, da sich der 20v8 nur in der Anzahl der Ein- und Ausgänge unterscheidet.) Auf der linken Seite sind die Eingänge zu sehen (alle Numerierungen entsprechen den Pinnummern am IC). Diese gehen sowohl normal als auch invertiert auf Spalten in der Logik-Matrix. Bei acht Eingängen erhält man also 16 Spalten. An den restlichen 16 Spalten sind die Rückführungen der Ausgänge (ebenfals normal und invertiert) angeschlossen, so daß man insgesammt 32 Signale zur Verfügung hat. Im unprogrammierten Zustand besteht keine Verbindung zwischen den 32 Spalten und den 64 Zeilen, d.h. an den Ausgängen kommt kein Signal an. Beim Programmieren werden nun Verbindungen hergestellt. Jede Verbindung entspricht einem UND, d.h. werden die Verbindungen so gebrannt wie im Bild 4 zu sehen (*: Verbindung vorhanden), so wird Ausgang A dann

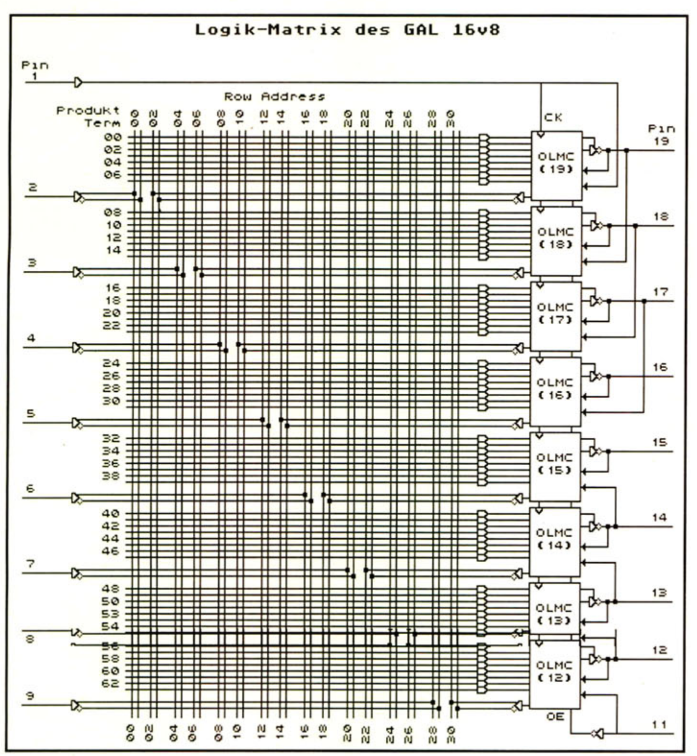


Bild 2: Interner Aufbau des GAL 16v8

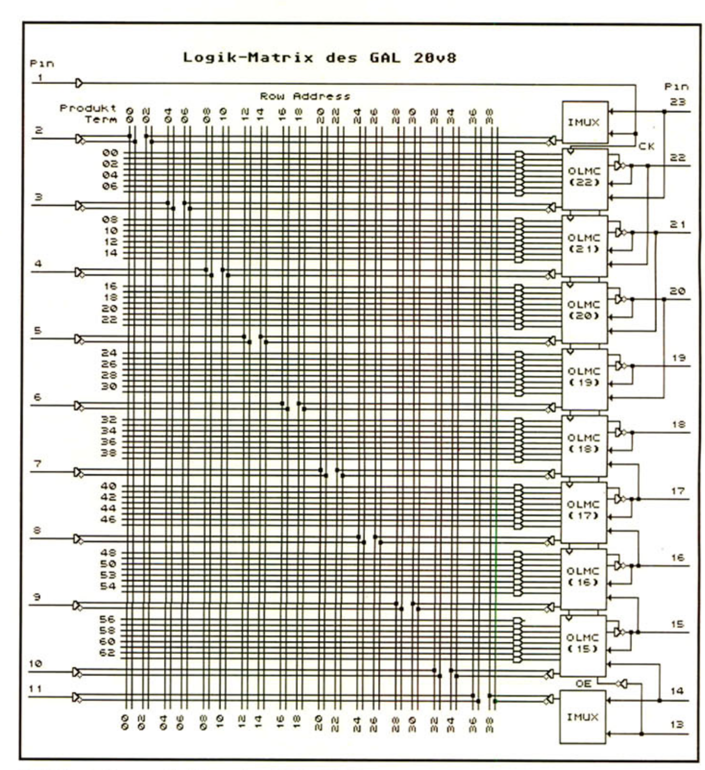


Bild 3: Interner Aufbau des GAL 20v8

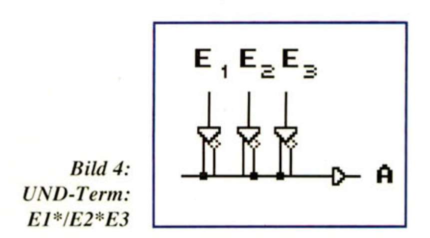


Bild 4: UND-Term: E1*/E2*E3

high, wenn Eingang 1 high, Eingang 2 low und Eingang 3 high ist (A = E1*/E2*E3). Jeweils acht Zeilen gehen ODER-verknüpft auf einen Ausgang, d.h. der Ausgang wird dann high, wenn mindestens eine der acht UND-Terme zutrifft (Aha! Jetzt haben wir ja, was wir für unser Beispiel brauchen!). Unsere Logikgleichung zur Erzeugung des Chip-Select-Signals sieht man in Bild 5. Der erste UND-Term wird durch die oberste und der zweite UND-Term durch die nächste Zeile realisiert. Um die ODER-Verknüpfung brauchen wir uns gar nicht weiter zu kümmern, da das der Baustein durch die Zusammenfassung der acht Zeilen von selbst macht. Doch dies ist noch lange nicht alles, was ein GAL kann. Kommen wir zum interessantesten Teil der *OUTPUT LOGIC MACROCELL* (OLMC, die Kästchen rechts in Bild 1 und 2), zu sehen in Bild 6. Drei Bits (SYN, AC0, AC1(n)) bestimmen, welchen Zustand der Ausgang haben soll:

- 'normaler' Ausgang (SYN = 1, AC0 = 0, AC1(n) = 0; Bild 8)
- 3-State-Ausgang mit Freigabe durch einen Produktterm und Rückführung (von den acht Zeilen werden nur die unteren sieben ODER-verknüpft und die oberste bestimmt, ob am Ausgang das Ergebnis erscheinen soll oder nicht. SYN = 0 oder 1, AC0 = 1, AC1(n) = 1; Bild 9)
- 3-State-Ausgang mit Register, Freigabe und Rückführung (erst wenn an Pin 1 (CK) ein Signal (Clockimpuls für Register) und an Pin 11 (/OE) kein Signal (inverser Freigabeanschluß) anliegen, erscheint am Ausgang das Ergebnis. SYN = 0, AC0 = 1, AC1(n) = 0; Bild 10)
- Eingang (SYN = 1, AC0 = 0, AC1(n) = 1; Bild 11)

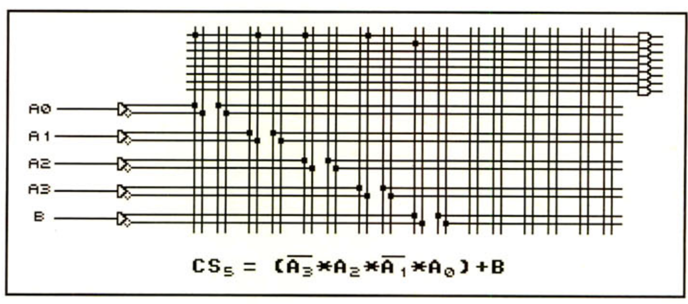


Bild 5: Unser Beispiel im GAL

Durch den letzten Modus besteht die Möglichkeit, mehr als acht Eingänge zu erhalten, z.B. ist ein Baustein mit 15 Eingängen und nur einem Ausgang programmierbar. Zusätzlich läßt sich mittels des XOR-Bits für jeden Ausgang einzeln bestimmen, ob er normal oder invers sein soll. Wenn wir für unser Beispiel einen inversen Ausgang mit Register benötigen, lauten die Bits: XOR(19) = 0, SYN = 0, AC0 = 1, AC1(19) = 0. Nachdem wir nun wissen, was in das GAL programmiert werden muß, fehlt uns nur noch das Wie.

Die Programmierung

Die Pinbelegung der GALs während der Programmierung ist in Bild 12 wiedergegeben. Um den Baustein in diesen Modus zu bringen, müssen am Pin 2 (Edit) 16,5 V anliegen. Nun kann das GAL gelesen, gelöscht und programmiert werden. Das GAL ist intern in 64 Reihen aufgeteilt

Reihe 60:
Architecture Control Word

Bedeutung der Bits	Produkttermfreigabe	XOR(n)	SYN	AC1(n)	AC0	XOR(n)	Produkttermfreigabe
beim GAL 16v8	PT63...PT32	12...15		12...19		16...19	PT31...PT0
beim GAL 20v8	PT63...PT32	15...18		15...22		19...22	PT31...PT0
Bit	81 50	49 46	45	44 37	36	35 32	31 0

Bild 6: Aufbau des Architecture Control Word

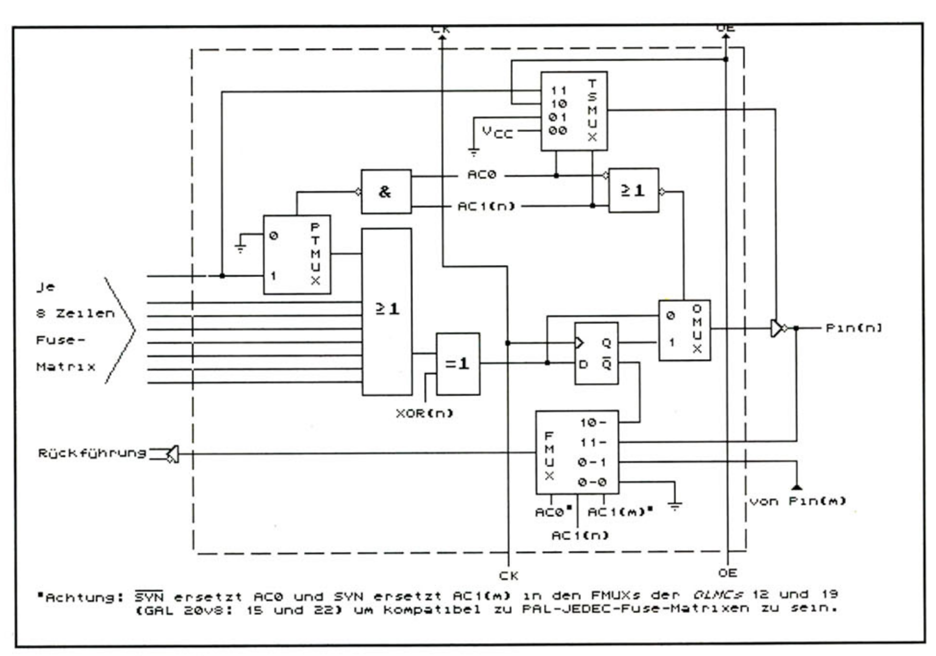


Bild 7: OLMC komplett

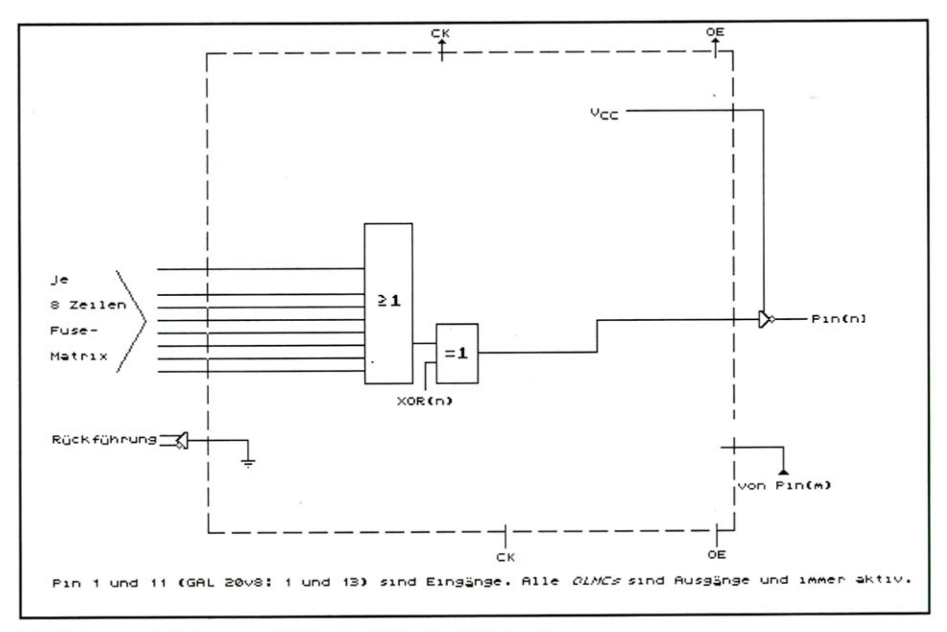


Bild 8: 'normaler' Ausgang (SYN = 1, AC0 = 0, AC1(n) = 0)

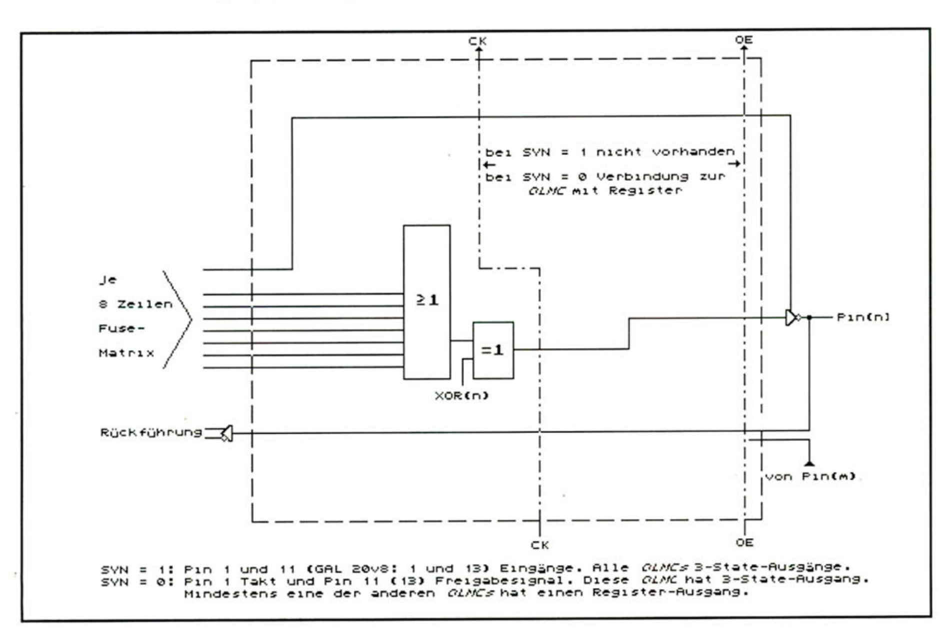


Bild 9: 3-State-Ausgang mit Freigabe und Rückführung (SYN = 0 oder 1, AC0 = 1, AC1(n) = 1)

(Bild 13). In den Reihen 0 bis 31 befindet sich die *Fuse*-Matrix. Diese kann direkt aus der Logik-Matrix gewonnen werden. Sie muß nur um 900 nach rechts gedreht werden. In der Reihe 32 stehen dem Benutzer 64 Bit für eigene Zwecke (z.B. Versionsnummer) zur Verfügung; diese Daten bleiben auch dann lesbar, wenn die *Security Cell* gebrannt wurde. Die Reihen 33 bis 59 sind beim GAL 16v8 nicht benutzt. In der Reihe 60 steht das *Architecture Control Word*, das 82 Bit lang ist (zum *ACW* später mehr). Wenn die Reihe 61 (*Security Cell*) programmiert wurde, kann die Fuse-Matrix (0-31) nicht mehr ausgelesen werden. Dies ist ein hardwaremäßiger Schutz, der sich nicht mehr rückgängig machen läßt (Kopierschutz!). Ein Beschreiben der Reihe 63 führt zur vollständigen Löschung des GALs (*Bulk Erase*). Die Reihen werden seriell über ein Schieberegister beschrieben und gelesen:

1.) Pin P, /V auf high (GAL wird programmiert) oder low (GAL wird ausgelesen) legen

2.) Mit den Pins RAG0 bis RAG5 die Reihe auswählen

3.) Lesen:
- mit einem /STR-Impuls das Schieberegister laden
- die Daten mit dem Takt SCLK an SDOUT auslesen

Programmieren:
- Beschreiben des Registers mit Daten an SDIN und Takt an SCLK
- Beschreiben der Reihe mit einem 10 ms /STR-Impuls

4.) weiter bei 2.), bis alle gewünschten Reihen gelesen oder programmiert sind

Im oben angesprochenen *Architecture Control Word*, dessen Aufbau im Bild 6 zu sehen ist, befinden sich die Angaben über den Zustand der Ausgangszellen. Außerdem gibt es dort 64 Bits, mit denen der Benutzter angeben kann, welche Produktterme er benutzen will. Wird das entsprechende Bit auf 0 gesetzt, so hat diese Reihe keinen Einfluß auf die Ausgänge. Dies wurde Vorgesehen, um eine Kompatibilität zu den PALs zu haben. PALs sind eine andere Gattung von programmierbaren Logik-Bausteinen. Sie haben gegenüber den GALs jedoch mehrere Nachteile:

- feste Ausgangszelle (d.h. die Polarität der Ausgangszelle und die Anzahl der Ein- und Ausgänge stehen beim Kauf des Bausteins bereits fest).

- höherer Stromverbrauch (beim GAL durch CMOS-Technik maximal 90 mA)

- nicht löschbar (d.h. wenn eine falsche Logik-Matrix ins PAL gebrannt wurde, oder sich diese später ändert, benötigt man ein neues PAL)

Zudem können die beiden GAL-Typen 16v8 und 20v8 jeweils 21 PAL-Typen emulieren (dazu mehr im nächsten Teil des Artikels). Als die GALs 1986 auf den Markt kamen, gab es bereits viele PAL-Typen. Da in vielen Fällen die Logik-Matrizen bereits in *Jedec*-Dateien (ein spezielles Format für die Logik-Matrix; näheres das nächste Mal) für PALs vorhanden waren, wählte *Lattice* diese Lösung. So kann auch ein bereits vorhandener PAL-Assembler (z.B. PALASM) weiter benutzt werden. Es muß nur an die PAL-Datei das entsprechende *ACW* angehängt werden. Soweit der erste Teil der Artikelreihe über GALs. Zugegeben: es war viel Theorie, aber ohne die Grundlagen kommt man nicht weiter und versteht auch die Programierung der GALs (z.B. als PAL-Emulation) nicht. Im nächsten Teil werde ich ausführlich auf das Thema GAL als PAL eingehen. Hierzu dann auch einige *ACWs* und mehr zur Jedec-Datei, wozu man sie braucht und wie man sie erstellt. Desweiteren gibt es einige Vergleiche zwischen in TTL-aufgebauter Logik und in GAL gebrannter. Im dritten Teil folgt dann als Selbstbauprojekt ein GAL-Programmiergerät für den ST inklusive Software, so daß man dann die Möglichkeit hat, selbst Erfahrungen mit diesen interessanten, vielseitigen Logikbausteinen zu sammeln.

Thomas Werner

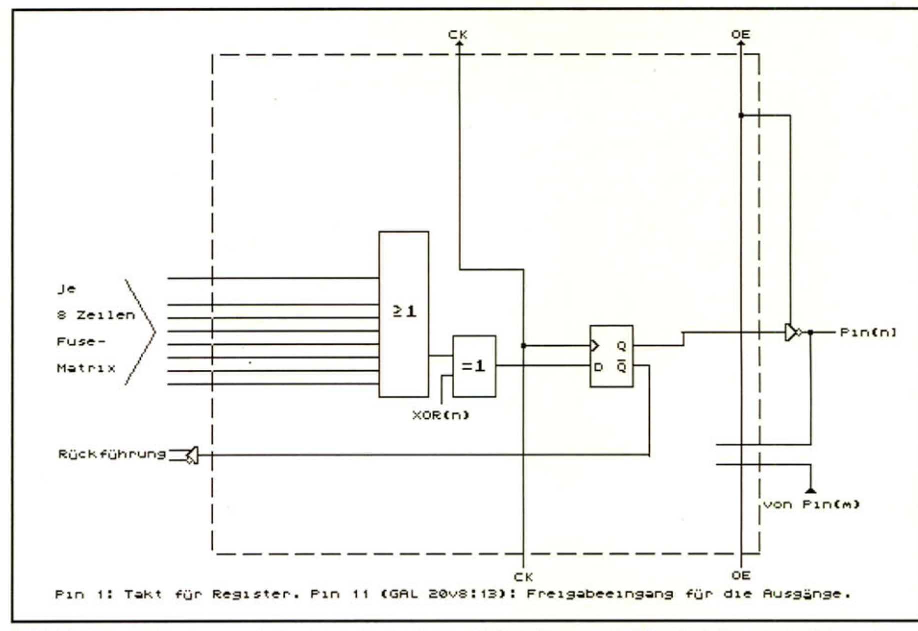


Bild 10: 3-State-Ausgang mit Register, Freigabe und Rückführung (SYN = 0, AC0 = 1, AC1(n) = 0)

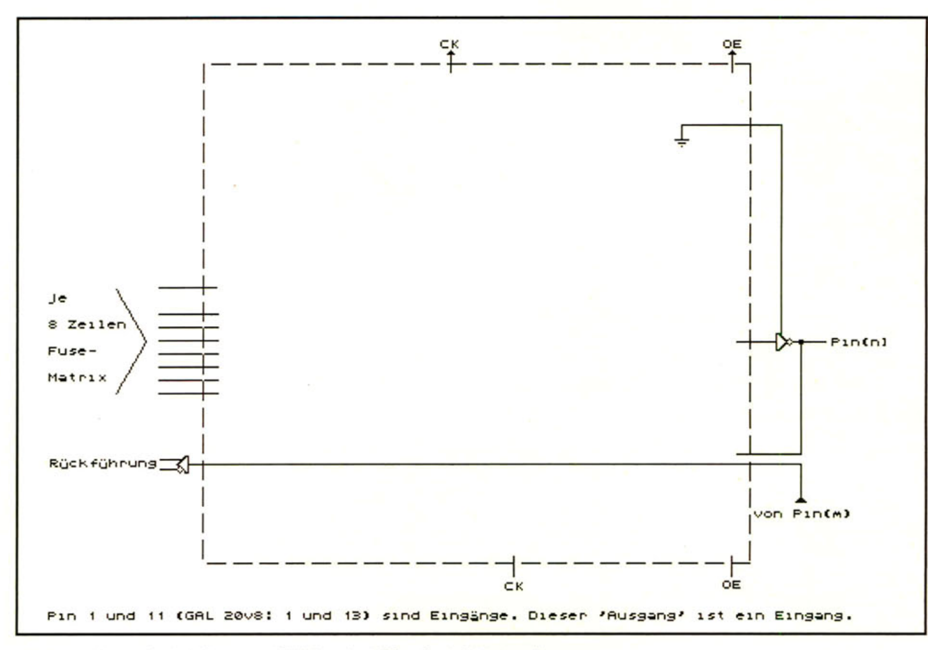


Bild 11: OLMC als Eingang (SYN = 1, AC0= 0, AC1(n) = 1)

Literatur:

Lattice: GAL Handbook
mc 1/88: Programmierbare Logikbausteine
TH Darmstadt: Anwendung integrierter Schaltungen, Skript

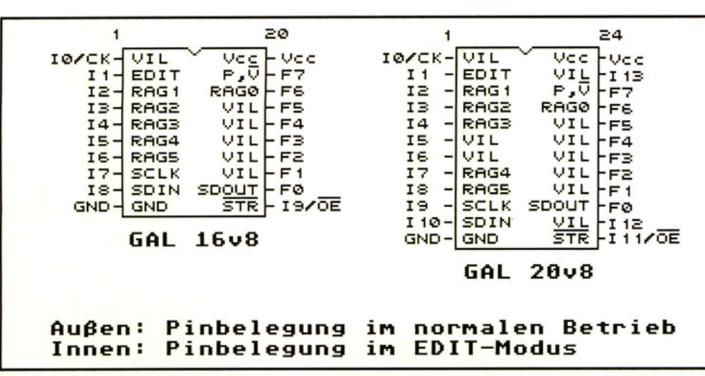


Bild 12: Pinbelegung der GALs 16v8 und 20v8 im Edit-Modus

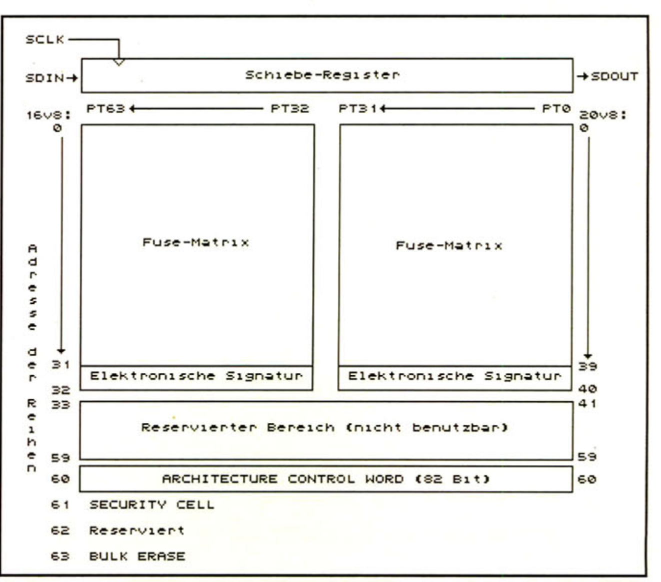


Bild 13: Interne Organisation

OMIKRON.

NEU 3.0

Basic



ATARI ST

Schneider/Steinmeier

kurz & klar OMIKRON BASIC

Ein fundiertes Nachschlagewerk

Heim-Verlag

DIE SPITZEN-BÜCHER ZUM NEUEN ATARI-BASIC



MERKMALE:

▶ Omikron-BASIC 3.0 – das neue Standard-BASIC für den ATARI ST – wird immer häufiger zur Erstellung von großen u. kleinen Programmen eingesetzt. ▶ Das Buch „Programmieren in Omikron-BASIC" zeigt an vielen Beispielen die Entwicklung von Programmen. Anhand zahlreicher dokumentierter Listings wird der Umgang mit den besonderen Elementen dieser BASIC-Version erläutert. ▶ In einem 1. Abschnitt werden wichtige Unter- u. Hilfsprogramme vorgestellt: Druckeranpassung, universelle Zahlensystemumwandlung, Feststellen der Existenz einer Diskettendatei, Programmierung grafischer Blockoperationen mit dem BITBLT-Befehl u.a.m. ▶ Ein weiteres Kapitel behandelt den Einsatz von GEM-Funktionen in Omikron-BASIC-Programmen: Der Umgang mit Fenstern, Dialogboxen, der Fileselectbox u.a.m. Auch die Verwendung der zum BASIC gehörenden GEMLIB wird behandelt. ▶ Um die Programmierung zu erleichtern, werden einige Hilfsprogramme gezeigt, die z. B. das Kopieren mehrerer Files vereinfachen oder eine Datei in DATA-Zeilen umwandeln. Auch ein Editor für Mauszeiger u. einige Füllmuster fehlen nicht. ▶ Die grafischen Fähigkeiten des Computers u. die entsprechenden Anweisungen des BASIC-Interpreters werden anhand der Entwicklung einer dreidimensionalen Grafik mit Rotation u. Projektion sowie einer Turtle-Grafik u. weiteren kleinen Beispielen demonstriert. ▶ 4 größere Anwendungen bieten die Gelegenheit, sich mit umfangreicheren Projekten vertraut zu machen. ▶ Für Schüler u. interessierte Laien sind einige Anwendungen aus dem mathematischen Bereich gedacht. ▶ Zur Abrundung u. Auflockerung enthält das Buch einige Spielprogramme.

INHALT:

▶ Druckeranpassung ▶ Feststellen der Existenz einer Datei ▶ Das Verwenden mehrerer Bildschirme ▶ Einbinden von Funktionen in laufende Programme ▶ GEM-Programmierung ▶ Arbeiten mit Fenstern, Dialogboxen u. Menüzeilen ▶ Punktgenaue Eingabe auf dem Bildschirm ▶ Komfortable Filenameneingabe mit Fileselect ▶ Programmierung der Alertboxen ▶ Hilfsprogramme ▶ Editor für Mauszeiger u. Füllmuster ▶ Kopieren von mehreren Files ▶ Dreidimensionale Darstellung von Funktionen ▶ Turtle-Grafik ▶ Adreßverwaltung ▶ Ermitteln von Mittelwerten, Varianz u. Standardabweichungen ▶ Integration nach Simpson ▶ Spiele

ca. 400 Seiten **DM 49,-** Best.-Nr. B-411

Diskette mit allen abgedruckten Programmen **DM 39,-** Best.-Nr. D-431

INHALT:

ATARI hat sich entschlossen, endlich eine wirklich leistungsfähige Programmiersprache mit den Rechnern der ST-Serie auszuliefern. Daß die Wahl gerade auf Omikron-BASIC fiel, ist kein Wunder, denn diese Sprache ist nicht nur besonders einfach zu erlernen, sondern stellt zugleich einen Leistungsumfang zur Verfügung, der selbst das Schreiben professioneller Anwendungen erlaubt. Um die über 200 Befehle mit ihren zahlreichen Parametern nutzen zu können, ist eine alphabetische Übersicht der Kommandos und ihrer Möglichkeiten unverzichtbar, und genau dies liefert „Kurz & Klar Omikron-BASIC 3.0".

Sie erfahren, mit welchen Anweisungen man Matrizen multipliziert, invertiert oder eine Determinante ermittelt, wie man Linien und Kreise zeichnet oder mit welchem Befehl man eine Alertbox auf den Bildschirm bringt. Außer der reinen Befehlsübersicht, die an sich schon eine enorme Hilfe bei der Programmentwicklung darstellt, geben die zahlreichen Anhänge Auskunft über die Bedeutung der Modi des BITBLT-Befehls, die ASCII-Zeichen des ST's oder die Zuordnung der Tasten zu den Scan-Codes der INKEY$-Funktion. Auch die VT52-Codes werden in einem Anhang behandelt.

Wenn man einen Befehl aus einem bestimmten Anwendungsgebiet sucht, hilft der nach Anwendungen geordnete Befehlsindex weiter. Dem Besitzer eines Compilers, der für das Erstellen eigenständiger Programme erforderlich ist, hilft ein Abshcnitt über die Compilerdirektiven und andere Besonderheiten weiter.

Die neueste Auflage dieses praktischen Hardcover-gebundenen Nachschlagewerkes berücksichtigt selbstverständlich die neue Version 3.0 des Omikron-BASICs.

Über 200 Seiten **DM 29,-** Best.-Nr. B-412

MERKMALE:

▶ Omikron-BASIC 3.0 ist der neue BASIC-Standard für den ATARI ST. Das vielfach bewährte „große Omikron-BASIC 3.0-Buch" gibt es nun in einer neuen Auflage, die alle Neuheiten berücksichtigt. ▶ Das Buch stellt einen leicht verständlichen Einstieg für den Anfänger in die Programmierung einer der leistungsfähigsten Sprachen für den ATARI ST dar. ▶ Jeder, der sich ernsthaft mit den Möglichkeiten dieser Programmiersprache beschäftigen möchte, benötigt dieses fundierte Lehrbuch. Aber auch der Umsteiger von einem anderen BASIC-Dialekt findet hier alle notwendigen Informationen, um mit Omikron-BASIC optimal arbeiten zu können. ▶ Dem Einsteiger bietet „Das große Omikron-BASIC 3.0-Buch" eine systematische und leicht verständliche Einführung, die von den Schleifen- und Programmstrukturen über die unterschiedlichen Variablentypen und die Arbeit mit Feldern bis zu den numerischen und Stringfunktionen reicht. Weiterhin findet der Leser Hinweise zum Umgang mit Unterprogrammen und Prozeduren. Aufbauend auf diesen Teil ist die Programmierung der Multitasking-Fähigkeiten anhand eines Drucker-Spoolers sowie der Einsatz von die Arbeit erleichternden abstrakten Datentypen erklärt. Weitere Abschnitte beschäftigen sich mit Dateien sowie mit der Programmierung von Grafik und Betriebssystemfunktionen. Auf diese Weise wird dem Neuling vom ersten Einzeller bis zu komplexen Programmen die Arbeit mit diesem komfortablen BASIC-Dialekt nahegebracht. ▶ Viele Beispielprogramme, die teilweise, wie etwa ein Fakturierungsprogramm, explizit entwickelt werden, runden das Buch ab. Damit die einzelnen Programme verständlicher werden, sind sie, wo nötig, durch Ablaufdiagramme ergänzt. Um nicht alle Listings abtippen zu müssen, liegt dem Buch eine Diskette mit allen Programmen bei. ▶ Aber auch, wenn man BASIC bereits beherrscht, ist das Buch durch seine zahlreichen Anhänge, die unter anderem eine ASCII-Tabelle, eine Übersicht der Füllmuster und BITBLT-Modi sowie ein Verzeichnis der VT52-Codes und einen Index enthalten, als Nachschlagewerk wertvoll. Zu diesem Zweck wurde auch eine vollständige alphabetisch sortierte Kurzübersicht der Befehle integriert.

INHALT:

▶ Erklärung der Schleifen- und Programmstrukturen – *Primzahlenberechnung – Zahlenraten* ▶ Variablentypen und Arrays – *Sieb des Eratosthenes – Adreßeingabe* ▶ Numerische und Stringfunktionen – *Alle trigonometrische Funktionen wie Sinus, Cosinus etc. – Die Zufallsfunktion* ▶ Unterprogramme und Prozeduren – *Rekursive Prozeduraufrufe – Suche des Ausgangs aus einem Labyrinth* ▶ Grafikprogrammierung – *Grafische Grundelemente wie Linie, Kreis, Rechteck – Blockoperationen – Flimmerfreie Animation durch Einsatz mehrerer Bildschirme – Erzeugen eines Balkendiagramms* ▶ Multitasking in Omikron-BASIC – *Druckerspooler* ▶ Programmierung von Abstrakten Datentypen – *Die Datenstruktur „Schlange" – Verkettete Listen* ▶ Dateiverwaltung – *Programmierung von sequentiellen Dateien – Arbeiten mit Random-Access-Dateien* ▶ Programmprojekt Fakturierung – *Ausführliche Entwicklung eines Programms auf ca. 30 Seiten* ▶ Betriebssystemprogrammierung – *Einsatz von Alertboxen und der Fileselectbox – Benutzen von Pulldown-Menüs, Fenstern und Dialogboxen – Programmierung einer Druckeranpassung in BASIC* ▶ Programmentwicklung und Debugging – *Fehlersuche und -beseitigung* ▶ Sammlung von Beispielprogrammen – *Ausgabe eines Diskettenverzeichnisses – Backup-Programm für die Harddisk – Turtle-Grafik in Omikron-BASIC – Adreßverwaltung* ▶ Alphabetische Befehlsübersicht

Hardcover über 400 Seiten einschl. Programm-Diskette **DM 59,–** Best.-Nr. B-413

Preise sind unverbindlich empfohlene Verkaufspreise

BESTELL-COUPON

an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Ich bestelle: _____ St. Das große OMIKRON-BASIC-BUCH (incl. Programm-Diskette) á DM 59,-

_____ St. Programmieren in OMIKRON-BASIC á DM 49,-

_____ St. PROGRAMMDISKETTE zum Buch á DM 39,-

_____ St. Kurz & Klar Nachschlagewerk OMIKRON-BASIC á DM 29,-

zzgl. DM 5,- Versandkosten (unabhängig von bestellter Stückzahl)

☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ________________

Straße, Hausnr. ________________

PLZ, Ort ________________

Benutzen Sie auch die in ST COMPUTER vorhandene Bestellkarte.

Schweiz
DataTrade AG
Langstr. 94
CH - 8021 Zürich

Österreich
Haider
Computer + Peripherie
Granzer Str. 63
A-2700 Wiener Neustadt

Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 0 61 51 - 5 60 57



„Please Let Me Introduce Myself"

Ich heiße Sie willkommen zu unserer neuen DFÜ-Ecke. Hier werden wir Ihnen Neuigkeiten in der "Szene", Grundlagen (wie XModem), Informationen und neue Mailboxen vorstellen. Sicherlich werden nicht alle von Ihnen wissen, was man mit der Datenfernübertragung alles machen kann. Aus diesem Grunde werde ich Ihnen in dieser ersten Folge erst einmal eine kleine Einführung bieten, damit später auch alle interessierten Leser mitlesen können.

Die Datenfernübertragung ist prinzipiell die Umwandlung von Daten in Töne. Diese Töne können beispielsweise über das heimische Telefonnetz überspielt werden. Auf der Empfängerseite werden die Töne wieder zurückgewandelt in Daten und können so wieder verarbeitet werden. Welche Arten der DFÜ gibt es nun? Diese Frage läßt sich sehr vielfältig beantworten. Da existieren zum einen die...

... Mailboxen,

in denen Sie auch als Privatperson anrufen können, um Daten zu lesen und/oder zu senden. Eine Mailbox kann man sich vorstellen wie einen Raum mit vielen Pinwänden. Jede Pinwand hat eine andere Überschrift, z.B. "Allgemeines", "Umweltschutz", "Politik" etc. An diesen Wänden hängen nun viele kleine Zettelchen, die die anderen Raumbenutzer geschrieben haben. Sie können diese Zettel lesen, aber auch eigene Zettel an die Wand hängen.

Bevor Sie jedoch überhaupt in den Raum gelangen, müssen Sie sich auf einer "Raumbenutzungsliste" eintragen, die vor der Tür hängt. Dadurch bekommen Sie einen eigenen Briefkasten, in dem Sie persönliche Post von den anderen Benutzern bekommen können. Wenn Sie den Raum wieder verlassen, geben Sie eine bestimmte Menge Geld an den Türwächter "Grün-Dollar". So sind Mailboxen aufgebaut. Allerdings bestehen die Pinwände hier aus sogenannten "Brettern", und die Raumbenutzer nennen sich "User". Die Zettel entsprechen den einzelnen Nachrichten in den Brettern, während der Türwächter Oberpostminister Schwarz-Schilling ist.

Es gibt verschiedene Arten von Mailboxen. Die einen sind kostenlos, die anderen kosten eine bestimmte Menge Geld pro Monat. Größere Unterschiede gibt es jedoch bei der Bedienung der Mailboxen: Verschiedene Menüsysteme machen es dem Einsteiger nicht leicht, sich zurechzufinden; deshalb werde ich Ihnen die wichtigsten Mailboxoberflächen näherbringen:

Menüsysteme

IMCA/GEONET nennt sich das (momentan) beliebteste System. Bei dieser Oberfläche müssen Sie alle Befehle als klarschriftliche, deutsche oder englische Befehle eingeben. Die Mailboxen, die dieses System benutzen, sind meist sehr komfortabel, weil durch die klarschriftlichen Befehle die Bedienung recht einfach wird. Eine Übersicht der wichtigsten Befehle (hier: *MagicBOX/MagicBOX ST*) finden Sie in Bild 1.

File- oder Zahlensystem nennt sich das andere, (noch) weit verbreitete System. Bei dieser Oberfläche werden alle Befehle durch Zahlen repräsentiert. Für den Einsteiger ist dies ideal, weil er sich nur drei "Befehle" merken muß. Mit der Zahl an sich (z.B. 280) wählt man ein File an, etwa "Allgemeines". Hat man das gewünschte File angewählt, muß man "01" zum Schreiben oder "02" zum Lesen eingeben. Weiß man gleich, daß man im File 280 lesen möchte, kann man auch die Zahlen addieren. Gibt man dann beispielsweise "282" ein, wird ins File 280 gesprungen und sofort gelesen. Hieraus ist zu erkennen, daß dieses System zwar wenig komfortabel ist, jedoch sehr einfach zu erlernen.

Mischsysteme sind auch noch im Umlauf. Hierbei können Sie beispielsweise "R 280" eingeben, um File 280 zu lesen (R="Read"), oder "S", um durch neue Nachrichten zu scannen. Leider kann ich Ihnen zu diesem System keine weiteren Erklärungen geben, da diese Programme sich so stark unterscheiden, daß die Befehle nur für eine bestimmte Mailbox gelten würden.

?		gibt eine Befehlsübersicht aus
Brett	*	zeigt eine Brettübersicht
Brett	**	zeigt eine ausführliche Brettübersicht
Brett	<name>	wählt das Brett <name> an
Hilfe		gibt einen Hilfstext aus
Hilfe	<befehl>	gibt den Hilfstext zum Befehl <befehl> aus
Inhalt		zeigt den neuen Inhalt eines Bretts an
Inhalt	*	zeigt den gesamten Inhalt eines Bretts an
Inhalt	x-	zeigt den Inhalt ab Nachricht x an
Inhalt	-x	zeigt den Inhalt bis Nachricht x an
Inhalt	x-y	zeigt den Inhalt von Nachricht x bis y an
Inhalt	<name>	zeigt alle Nachrichten von User <name> an
Lesen		liest alle neuen Nachrichten
Lesen	*	liest alle Nachrichten
Lesen	x	liest Nachricht Nummer x
Lesen	x-	liest alle Nachrichten ab Nummer x
Lesen	-x	liest alle Nachrichten bis Nummer x
Lesen	x-y	liest alle Nachrichten von Nummer x bis y
Lesen	x,y,z	liest die Nachrichten x, y und z
Logoff		beendet die Verbindung
Senden		an das aktuelle Brett senden
Senden	<name>	an den User <user> senden
Senden	<brett>	an Brett <brett> senden

Bild 1: Die wichtigsten Befehle nach IMCA/GEONET (hier: MagicNET)

Vernetzung

Da Mailboxen in den letzten Jahren wie Pilze aus dem Boden geschossen sind, konnten nicht alle überleben. Aus diesem Grund haben sich viele Mailboxnetze entwickelt. In diesen Netzen sind viele Mailboxen zusammengeschlossen und tauschen allmorgendlich ihre neuen Daten aus, damit mehr Benutzer etwas davon haben. Leider entstehen momentan immer mehr neue Netze. Dies ist eine schlechte Sache, da wir in Deutschland lieber *EIN* großes Netz haben sollten als viele kleine. Einige Netzkoordinatoren stellen sich jedoch quer und möchten dies auf keinen Fall. Die Zukunft wird zeigen, wer in der Netzvielfalt überleben wird. Ich stelle Ihnen die wichtigsten Netze vor und gebe zu jedem meine subjektive (!) Meinung wieder:

Das *MagicNET* entstand 1987 durch *Ingo Richardt*, den man getrost einen Pionier der deutschen Mailboxszene nennen kann. In diesem Netz, an dem zur Zeit über 45 Systeme teilnehmen, existiert die größte Befehlsvielfalt mit über 80 komfortablen Befehlen. Immer mehr Mailboxen treten diesem Netz bei, weil sie die Benutzeroberfläche als die beste empfinden. Das MagicNET ist zum größten Teil in Nordrhein-Westfalen verbreitet, allerdings existieren auch Mailboxen in Mannheim und Hamburg. Im MagicNET ist die Möglichkeit gegeben, *"Point"* zu werden. Als Point erhält man alle Netznachrichten über den normalen Netztransfer in gepackter Form und kann sie in aller Ruhe zu Hause durchlesen, ohne Telefongebühren zahlen zu müssen. Dies ist bei keinem anderen deutschen Mailboxnetz möglich (nur im Fido-Net, das aber nicht deutsch ist). Ein weiterer großer Vorteil dieses Netzverbunds ist, daß die Sysops (*SYS*tem *OP*erator = Systembetreiber) sich bemühen, andere Netze in ihr eigenes mit aufzunehmen. Dafür existiert eine eigene Mailboxebene, *"Gate"* genannt. Hier tummeln sich momentan Nachrichten aus dem Zerberus- und dem Fidonet. Wenn Sie also eine große Nachrichtenvielfalt, verschiedene Netze in einem und eine gelungene Benutzeroberfläche suchen, wählen Sie das MagicNET. Übrigens: Dieses Netz läuft auch mit ATARI ST-Rechnern! Den *Netzplan* des MagicNET finden Sie in Bild 2.

Das *PCNet* existiert ebenfalls seit ca. 1987 und wurde auch von einem der Mailboxpioniere Deutschlands gegründet, von *Dirk Gazic*. Von ihm stammt auch das weithin bekannte Programm *"64sysop"* für den C64, das heute überall als Quellprogramm erhältlich ist und in vielen abgewandelten Formen sein Dasein fristet. Das PCNet kann momentan nur mit IBM-kompatiblen Rechnern betrieben werden, eine Umsetzung für den ST ist jedoch bereits in Arbeit. Als Benutzeroberfläche ist hier eine abgewandelte Form von IMCA/GEONET gewählt worden; dem Standard entspricht sie jedoch in keinem Fall. Auch die Befehlsvielfalt läßt in manchen Teilen zu wünschen übrig, ebenso die unansprechend gestaltete Benutzeroberfläche. Das PCNet ist ebenfalls im Großraum Nordrhein-Westfalen (Köln) vertreten. Nach einem Streit des Netzkoordinators mit dem Sysop der Foto-Box hat sich ein anderes Netz mit dem gleichen Programm gebildet, das *WWM-Netz*. Da dieses nur eine Absplitterung des PCNet ist, brauche ich darauf wohl nicht weiter einzugehen.

Das *ZerberusNet* ist das größte deutsche Netz. Weit über 70 Mailboxen nehmen daran teil. Leider läßt auch hier die Benutzeroberfläche etwas zu wünschen übrig, ist jedoch noch akzeptabel. Hier werden viele Möglichkeiten geboten, die das MagicNET nicht hat: *Batchfiles* können aufgerufen werden, Brett-Unterbretter (die nicht dem IMCA/GEONET-Standard entsprechen) können eingerichtet werden etc. Über das ZerberusNet kann auch *LINKS* (eine "linke" Vereinigung) erreicht werden. Auch hier ist der Vorteil, daß das Programm auch mit einem ATARI ST betrieben werden kann, leider existieren aber keine Points (s.o.), vielleicht ändert der Programmautor dies noch ab? Für die, die das Programm schon kennen, habe ich noch eine Information: Das Zerberus-Programm wird momentan in C umgeschrieben und dürfte damit das schnellste aller deutschen Programme werden. Als der Hit ist in das Programm eine Programmiersprache eingebaut, die vergleichbar mit dem *Commodore-*

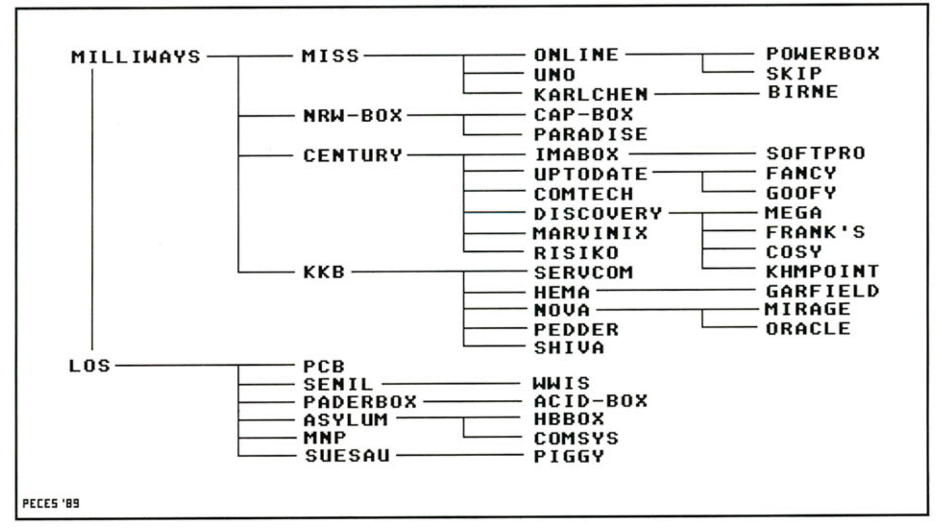


Bild 2

Basic V2 (C64) ist: Damit lassen sich beispielsweise Batchfiles in ungeahnter Qualität erstellen! Man darf gespannt sein. Das ZerberusNet ist über das gesamte Land verteilt und verfügt auch über Außenstellen in anderen Ländern.

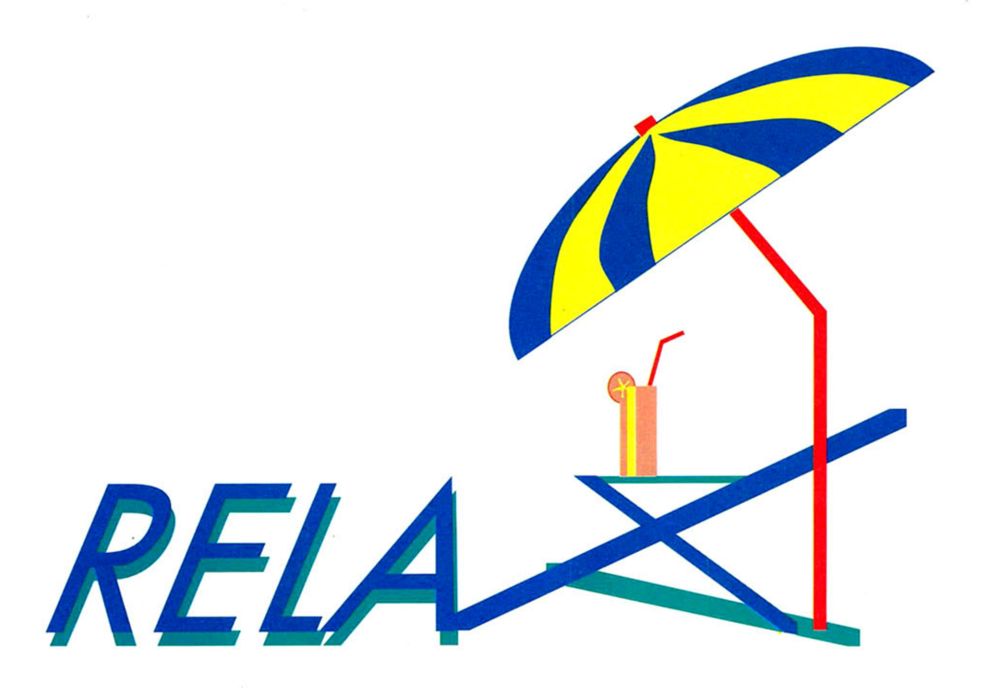
Weitere Netze sind das *AMNet II* (AMIGA), das seine Benutzeroberfläche dem MagicNET abgeguckt hat (man merkt es gleich), das *StarNet* (ATARI ST) mit einer keinem Standard entsprechenden Benutzeroberfläche, die zudem noch recht kompliziert zu überblicken ist, sowie das *FidoNet* (über 6000 Mailboxen in aller Herren Länder), welches aber keinem Anfänger zu empfehlen ist, da die Befehle hier doch recht kompliziert sind.

Andere DFÜ

Natürlich gibt es nicht nur Mailboxen, sondern auch noch andere Anwendungsbereiche der Datenfernübertragung. Bestes Beispiel hierfür ist der immer weiter verbreitete *Telefax*. Ein Telefaxgerät ist nichts weiter als ein Thermo-Transfer-Drucker, mit dem man seine Texte an einen anderen Thermo-Transfer-Drucker senden kann. Ganz so einfach ist es aber auch nicht: Eine Vorlage, meistens ein Schriftstück, wird in das Telefaxgerät gelegt. Dann wählt man die Nummer eines anderen Geräts an. Sobald eine Verbindung zustandegekommen ist, wird die Vorlage abgescannt, also digitalisiert. Digitalisierte Werte kann man sehr einfach in Töne umwandeln (Ton/kein Ton). Genau das geschieht auch: Die gescannten Bildpunkte werden in Töne umgewandelt und über die Telefonleitung geschickt. Auf der Empfängerseite werden sie wieder in Daten zurückgewandelt. Diese nimmt nun der empfangende Thermo-Transfer-Drucker in Empfang und spuckt zeilenweise die empfangene Seite aus. Diese Art der DFÜ ist zwar stark eingeschränkt, man kann sie aber immer noch DFÜ nennen. Sicherlich existieren noch weit mehr Anwendungsmöglichkeiten der DFÜ, auf die ich aber hier nicht eingehen kann.

Damit bin ich auch schon am Ende der ersten Folge der DFÜ-Ecke angelangt. Wenn Sie Fragen zu einem beliebigen Thema der DFÜ haben, schreiben Sie mir: Ich werde versuchen, alle Fragen entweder im Rahmen der DFÜ-Ecke oder privat zu beantworten. Sie können auch Wünsche äußern, was besprochen werden soll: Ich bin für alle Wünsche offen. Schreiben sie an

MAXON Computer GmbH
Redaktion ST-Computer
Stichwort: DFÜ-Ecke
Industriestraße 26
6236 Eschborn



Electronic Arts bietet jetzt eine Zusatzdiskette für den Spielknüller *Populous* an, *The Promised Lands*. Auf ihr sind weitere fünf neue Welten enthalten, die das Spiel erfreulicherweise stark ausbauen. Man sollte sich aber nicht nur auf altgewohnte Strategien verlassen, da man damit leicht einen Einbruch erleben kann. An neuen Welten gibt es eine Computerwelt namens The Bit Plains, ein Art Legowelt namens Blockland, ein Szenario zur Französischen Revolution und eins, das im Wilden Westen spielt und zuletzt noch ein Silly Land mit Riesen.

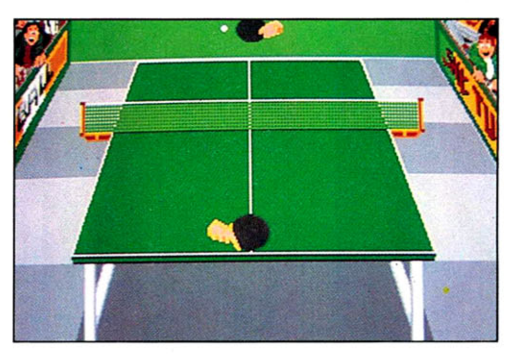


The Promised Lands

H*ewson* setzt nach dem schaurigen *Astaroth - Angel of Death* auf Roboter. In *Steel* steuern Sie eine fliegende Konservendose (sieht jedenfalls so aus!) durch Gänge eines Raumschiffs und zerstören per Feuerknopfdruck feindliche Blechkameraden. Hübsche Grafik! Aber wie so oft bei Hewson-Spielen, mangelt's an der Spielbarkeit. Hat man einen Roboter abgeschossen, rasen sofort zwei neue heran. Kein Wunder, daß die Energieanzeige schnell gegen Null strebt. So wars jedenfalls bei unserer Vorabversion. Hoffentlich ändert *Hewson* das noch!

I*ndiana Jones and the last Crusade* heißt das Such-, Sammel-, Kletter-, Hüpf-, Kampf- und Schießspielchen von *U.S Gold*. Ich glaube, Indy hätte den Programmierer ausgepeitscht... Fanatische Fans sollten auf das gleichnamige Adventure von Lucasfilm Games warten.

Kurz vor Redaktionsschluß erreichte uns *Turtle's Table Tennis* von *Starbyte*: Weltmeisterschaft, Weltrangliste, Ein- und Zweispieler-Modus, Einzel und Doppel, das sind die aufregensten Features dieses Tischtennisprogramms. Für ein kurzweiliges Ping Pong-Intermezzo ganz brauchbar, ansonsten bietet das Spiel bloß einen durchschnittlichen Unterhaltungswert.

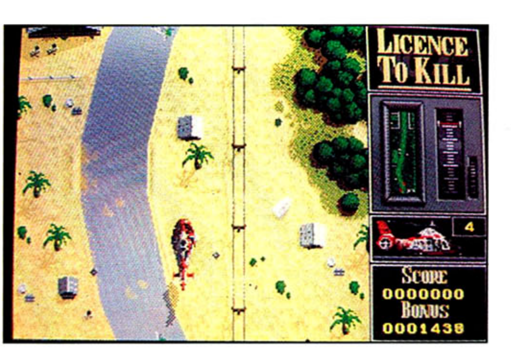




Turtle´s Table Tennis

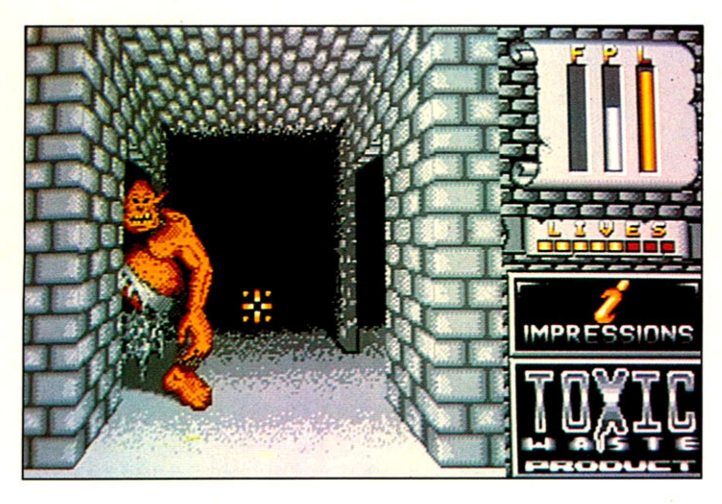
X*enon II* heißt das absolute Megaballerspiel. Tierisch viele Sprites, farbenprächtige Grafiken, beeindruckendes Scrolling und hohe Motivation machen das Programm zu einem Superknüller!

James Bond 007, *Domarks* Doppelnull, erlebt sein neustes Filmabenteuer *Lizenz zum Töten* auch am ATARI ST. Das Spiel beeindruckt durch viele verschieden gestaltete Levels: Bond fliegt Hubschrauber, Flugzeuge, schießt im Meer mit Harpunen auf Schwimmer von Wasserflugzeugen, liefert sich Schießereien mit Sanchez' Privatarmee und rammt Tanklastzüge. Für Freaks von Actionspielen ein heißer Spaß!





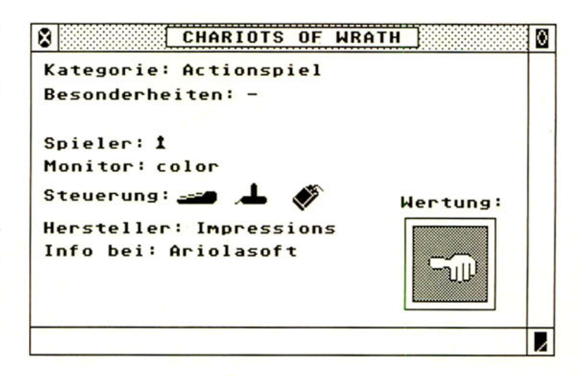
Lizenz zum Töten



Chariots of Wrath

Es ist mal wieder eine wunderhübsche Prinzessin entführt worden. Was geht Sie das an? Ganz einfach! Bei dem holden Mädel handelt es sich um Ihre Braut. Als stahlharter Ehrenmann schlagen Sie sich wagemutig durchs gesamte Königreich, um das liebe Mädchen zu befreien. Das geschieht etappenweise: Auf einer Karte des Reiches markiert ein Kreuz die gegenwärtige Position des Helden. Je nach Standort ist eine von vier Actionszenen zu absolvieren. Handlungsort der ersten Spielsequenz ist eine mittelalterliche Burg, in deren Verliesse riesige Schurken hinter den Säulen hervortreten. Mit einem Fadenkreuz zielt man auf die Biester und erledigt sie durch einen Druck auf den Feuerknopf. Durch diese Missetat erhält der Held Benzin. In einer anderen Szene hüpft der Retter über Plattformen und sammelt dabei blinkende Karos ein, die ebenfalls zu mehr Sprit verhelfen. Wozu braucht man bloß im Mittelalter Benzin? Diese Frage beantwortet sich von selbst, spielt man die dritte Szene. Hier steuert der Spieler ein Raumschiff durch einen dunklen Gang und knallt feindliche Objekte wie Totenköpfe und schleimende Würmer ab. Diese Szene paßt überhaupt nicht ins mittelalterliche Ambiente. Trotzdem macht sie am meisten Spaß. Das Ballerspielchen mit vertikalem Scrolling verfügt über viele spritzige Extrawaffen und eine schöne Grafik. Am Ende dieses Levels wartet ein riesiges Endmonster darauf, durch gezielte Schüsse erledigt zu werden. Hat man das geschafft, ist der Weg zum nächsten Spielabschnitt frei. Neben den drei beschriebenen Sequenzen gibt es noch eine vierte, in der man genau wie bei *"Arkanoid"* Steine mit Hilfe eines Schlägers und einer Kugel aus einer Mauer bricht. Alle Spielszenen wechseln sich stets in unterschiedlicher Reihenfolge ab. Einerseits schafft das englische Softwarehaus "Impressions" durch den Spielemix Abwechslung. Andererseits müssen bei allen Teilspielen Abstriche gemacht werden, da für jede einzelne Szene natürlich weniger Speicherplatz zur Verfügung steht, als wenn eine solche Szene ein eigenständiges Spiel wäre. Daher kommt es, daß jede Sequenz (bis auf das Ballerspiel) sehr primitiv gehalten ist. Für ein schnelles Spielchen zwischendurch ist *"Chariots of Wrath"* geeignet. Doch wer ein Spiel kauft, der möchte länger etwas davon haben. Und das ist bei *"Impressions"* Spielmix nicht gegeben.

CBO


CHARIOTS OF WRATH

Kategorie: Actionspiel
Besonderheiten: -

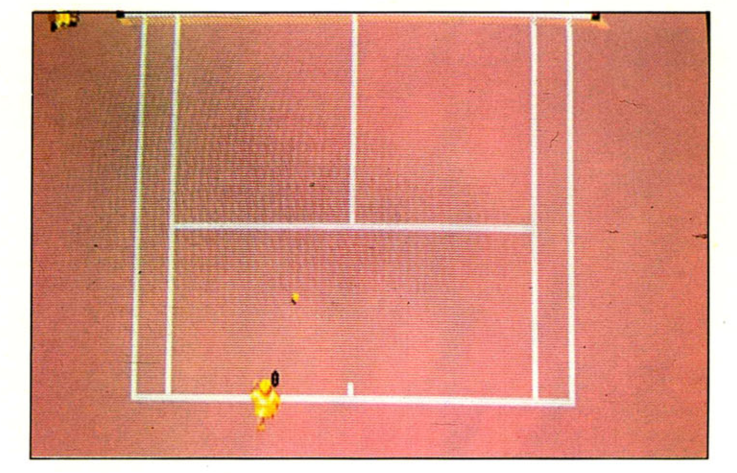
Spieler: 1
Monitor: color
Steuerung:
Hersteller: Impressions
Info bei: Ariolasoft

Wertung:

Passing Shot

Seit Boris und Steffi in Wimbledon abgeräumt haben, kommen auch Spieledesigner auf den Geschmack, Tennisspiele zu programmieren. Mittlerweile arbeiten drei Softwarehäuser an der Fertigstellung eines solchen Games: Blue Byte hackt "Great Courts" zusammen, Image Works macht "Passing Shot" und "Starbyte" entwickelt ebenfalls ein Tennisprogramm. Alle drei sollen noch vor Weihnachten auf den Markt kommen. Da will jeder schneller veröffentlichen als der andere; schließlich möchte man nicht wertvolle Marktanteile an den Konkurrenten verlieren. "Image Works" hat's als erster geschafft: "Passing Shot" ist schon Anfang Oktober in den Geschäften. In all der Hast wurde aus der Umsetzung zum gleichnamigen Sega-Automaten nur ein Durchschnittsspiel, bei dem Realitätsnähe und Spielspaß auf der Strecke bleiben. Daß man die Action aus der Vogelperspektive sieht, läßt sich verschmerzen. Fast ruckelfreies Scrolling, das sich stets mit dem Ball bewegt, und die dazu richtige Perspektive, bringen Schwung ins Spiel. Zu Frust führt allerdings, daß man auf dem Bildschirm nur den Ausschnitt des Tenniscourts sieht, in dem sich gerade der Ball befindet. Über den eigenen Tenniscrack hat man so oft keine Kontrolle, da er nicht immer im Bild zu sehen ist. Schmettert der Gegner einen Ball übers Netz und man hat seinen Spieler intuitiv irgendwohin gesteuert, läßt sich der Ball schwer retournieren. Da wird Tennis zum Glücksspiel! Ebenfalls nicht zufriedenstellend ist die Tatsache, daß man für einen Grand Slam-Tuniersieg in Frankreich, Australien oder den USA nur einen Satz zu gewinnen braucht. Positiv fielen dagegen die relativ starken Computergegner und die gut realisierten Schläge Top Spin, Lob und Slice auf. Auch der Perspektivenwechsel beim Aufschlag kam bei unserem Testerteam gut an. Doch die spielerischen Schwachpunkte dominieren. Deshalb würde ich Tennisfreaks empfehlen, auf die beiden anderen Programm von Blue Byte bzw. Starbyte zu warten. Vielleicht werden die besser?

CBO



PASSING SHOT

Kategorie: Tennisspiel
Besonderheiten: -

Spieler: 1...11
Monitor: color
Steuerung:
Hersteller: Image Works
Info bei: Ariolasoft

Wertung:



Kleine Sensationen gibt es gerade im PD-Bereich immer öfter, hier eine kurze Zusammenfassung: TeX, das Profi-Satzsystem, erscheint auf Pool-Diskette 2049 und 2050, MD-Software gibt einen Zeichensatz für Calamus frei (2045) und 12. Jahrhundert wurde erweitert (2042). Übrigens, wer die PD-Szene noch nicht kennt, sollte sie jetzt bei seinem PD-Pool-Händler anfordern.

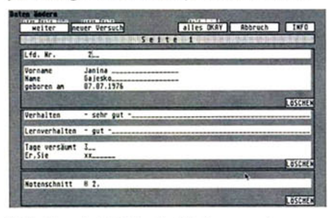
In der PD-Szene werden alle Programme der 2000er Serie vorgestellt. Außerdem finden Sie dort die aktuelle Liste der 'Top 1000' PD-Programme - nach Themengebieten sortiert, wie in der PD-Fibel. Nebenbei verraten wir Ihnen auch, warum unsere PD-Weihnachtsaktion bereits in einigen Wochen anläuft und was sich dabei alles abspielt.

Viel Spaß mit den neuen Programmen,

Die "23"

2041

Adressmanager 2.0 D, jetzt mit neuen Sortier-, Import/Export-, Ausgabe- und Listenfunktionen. Toll gemacht, von Jörg Trojan (s/w).
Formular 2.4 bedruckt Briefe, Postkarten, Überweisungen und Formulare aller Art. Genaues Positionieren aller Textfelder durch Millimeterangabe. 9- und 24-Nadel-Anpassung, Rechnen, Speichern (s/w).



Minitext 2.45: die kleine, praktische Textverarbeitung. Jetzt mit 22 Zeilen, neuer Dialogbox, Tastaturbelegung und schnellem Scrolling (s/w).

Fontmonitor ermittelt, welche Fonts in einem Signum-Dokument (SDO) verwendet wurden (s/w).
Suchsets 1.1 gibt eine alphabetische Liste der verwendeten Signum-Fonts aus (s/w).

2042

12. Jahrhundert 0.4: historische Simulation der Spitzenklassen. Jetzt mit 1-4 Spielern, Justiz, Festungsbau und Missionieren. Wirtschaft, Politik, Religion und Krieg fordern Ihr volles Engagement (s/w).



Celest: Strategiespiel für 1-4 Space-Cadets, bei dem Sie den Weltraum erforschen, Planeten erobern, Raumschiffe bauen und die Flotten der Gegner bekämpfen (f).
Profi-Liga verwaltet Spielergebnisse der Bundesliga, mit Stand vom 25.8.89. Alle Spieltage der Saison 89/90 wurden bereits eingegeben. Sie planen, simulieren und fiebern, bis der Meister feststeht (s/w).

Spieltag-Übersicht

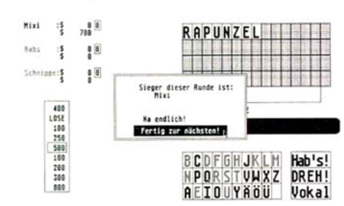
FC St.Pauli	- FC Homburg	1 : 1	Spieltag + 1
Bayer Leverkusen	- Karlsruher SC	1 : 1	
FC Bayern München	- 1.FC Köln	5 : 1	Spieltag - 1
1.FC Kaiserslautern	- Waldhof Mannheim	2 : 3	
Bayer Uerdingen	- Hamburger SV	5 : 2	
VfB Stuttgart	- Eintracht Frankfurt	1 : 1	Schrittweite
Borussia Dortmund	- 1.FC Nürnberg	2 : 1	
Werder Bremen	- Bor. Mönchengladbach	0 : 0	Abbruch
Fortuna Düsseldorf	- VfL Bochum	2 : 2	

Spieltag: 5

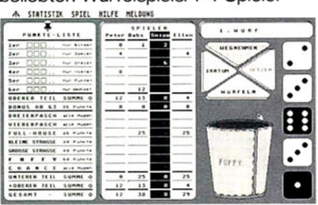
Fussball-ST 1.61 versetzt Sie in die Rolle eines Managers. Spielertransfer, Finanzplanung und Stadionausbau gehören zu Ihren wichtigsten Aufgaben, wenn die Saison ein Erfolg werden soll (s/w). 1-4 Spieler

2043

Glücksrad: das Spiel zur TV-Serie. Nettes Ratespiel, bei dem 2-3 Spieler um das große Geld kämpfen.
Quiz kennen Sie ebenfalls aus dem Fernsehen. Für 1-7 Spieler geht es hier um 96 Fragen und den großen Preis (f & s/w).

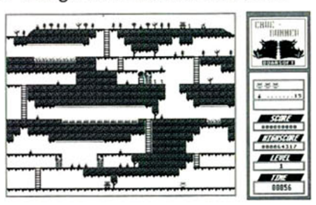


Lotte '6 aus 49' Vollsystem-Simulation für angehende Millionäre. Gewinnquoten, Kontostand, automatische Ziehung und mehr...
Fuffy 2.0, die neue Version des beliebten Würfelspiels. 1-4 Spieler



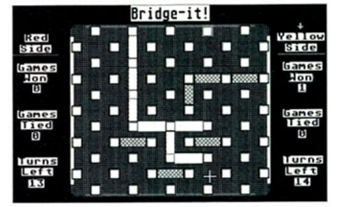
Single-Skat, das Spiel für Einen allein (s/w). Spielstark.

Cave-Runner klettert, rennt, sammelt und kämpft sich durch das Labyrinth, Level für Level. Dafür hat er lässige 45 Sekunden Zeit (s/w).

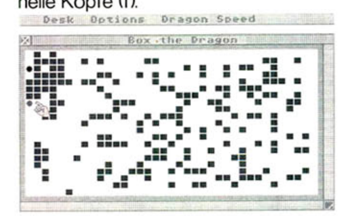


2044

Thonatos: grafisch sehr schön aufgemachtes 'Wurmspiel', bei dem die 2 Spieler versuchen, den Wurm des Gegners zur Kollision zu bringen.
Bridge-it: Strategiespiel für 1-2 Brückenbauer. Sieger ist, wer sein Bauwerk als Erster vollendet (f).



Orions Run: Weltraum-Action der Spitzenklasse. Bomb and Shoot everything in sight (f).
Ripcord: lustiges Sportspiel, Fallschirm-Zielspringen (f).
Two-Game: Mastermind und 'Fang den Drachen' - zwei Taktikspiele für helle Köpfe (f).



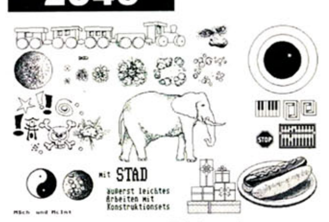
Dragon: der Drache schlägt zurück. Fliehen Sie aus dem dunklen Labyrinth, bevor der Drache Sie verspeist (f).
Black Jack: Kartenspiel (f).
Kniffel: Würfelspiel mit sehr guter Grafik. 1-7 Spieler (f).

(f) = Farbe
(s/w) = monochrom

2045

ASchreib: Ein Zeichensatz für Calamus DTP. *In die PD gegeben von MD-Software. ABCDEFG-HIJKLMNOPQRSTUVWXYZ-abcdefghijklmnopqrstuvwxyzäöü-1234567890. Schriftprobe.*
Calamus-Demoversion, anhand der Sie die vielseitigen Funktionen des DTP-Programms, ausprobieren können. Drucken ist möglich, dabei wird aber der Hinweis 'Demoversion' diagonal über die Seite gedruckt.

2046



Clip-Art: über 60 Bilder mit hunderten von Clips (s/w).
IDL Show: Bildverwaltung ganz einfach gemacht (s/w).
Show-Bild, die Dia-Show (s/w).
STAD Dia-Show/Animation (s/w).



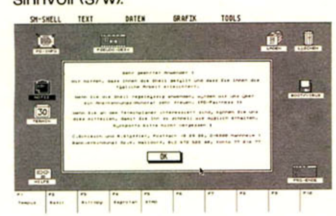
2047

EINSTELLEN ACTION FILE TEXTER

Warten auf Maus- oder Tastendruck
Alertbox konstruieren
Verzögerung um eine bestimmte Zeit
Mausform editieren
Menuleiste erstellen
Setcolor" (zum Invertieren des Bildschirms)
Kreis zeichnen
Ellipse zeichnen
Befehl eingeben
Zurück zum Menü

Texter 2: GFA-Tool, mit dem Grafik-Befehle per Maus erzeugt und als Merge-File gespeichert werden (s/w).
Crossref 2.1 für GFA-Basic.
GFA-Edit 1.9, neue Version des schnellen Editors (s/w).
No-CLS entfernt 'CLS' aus GFA-Basic Programmen.

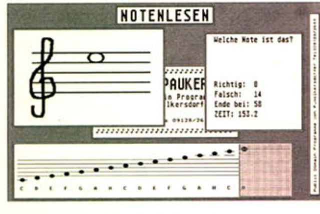
Bootutility 2.0, bringt Fastload, 2ms Seekrate, Zeit/Datum, Viren-Schutz, Software-Reset und mehr. **SM-Shell**, das neue Desktop für Festplattenbesitzer. Bis zu 45 Programm können blitzschnell über Pulldown-Menüs aufgerufen werden, 10 weitere über die F- Tasten. Sehr sinnvoll (s/w).



Virendetektor 2.8 überprüft Disketten, Programme und Partitionen auf Bootsektor- und Linkvirenbefall. **Hard-Inf** läßt den Anwender unter mehreren DESKTOP.INF das Gewünschte auswählen. **HD-Scan**: schnelles Suchen, Markieren, Kopieren oder Löschen von Dateien auf Disk und Festplatte.

2048

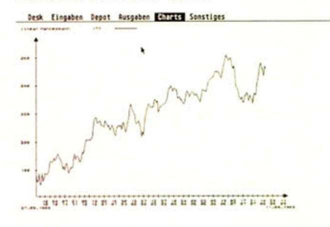
Midi-Pauker 1.0 Lernprogramm zum Notenlesen und Anzeigen von Noten die auf dem Keyboard gespielt werden. Darstellung der Noten in Violin- oder Baß- schlüssel.

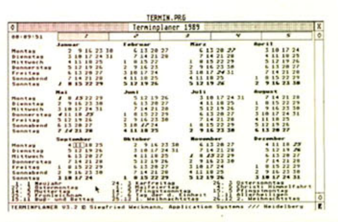


Kaufmännische Berechnungen, vom Kleinkredit über Eigenbau-



Kalkulation bis hin zur Break-Even-Analyse (s/w). **Vermessungsprogramm** nach Rene Descartes (s/w). **KC&D**: Kurse, Charts & Depot. Tageskurse, 200-Tage Schnitt, Minimum, Maximum, %Steigerung, Varianz und mehr (s/w).





Terminplaner 3.2: die neue Version sorgt dafür, daß Sie Ihre Termine im Griff haben. Adressverwaltung integriert, verschiedene Kalender, sehr umfangreiches Handbuch, läuft auch als Accessory.

2049

2 Disketten: 2049_A & B

TeX: Satzprogramm der Spitzenklasse - Shareware von Stefan Lindner. **Keine** Demo, sondern die neue Vollversion (1MB RAM, zwei Laufwerke, bzw. Festplatte).

2050

TeX-Fonts und Druckertreiber passend zu Disk 2049. Bitte das gewünschte Treiberpaket im Bestellcoupon ankreuzen (TeX - C bis G).

C - 9-Nadel
D - NEC P6/P7 (kompat.) 180dpi
E - NEC P6/P7 (kompat.) 360dpi
F - Epson LQ und kompatible
G - ATARI Laserdrucker SLM 804
(C,D je 1 Disk, E-G je 2 Disks)

PD-Szene schon gelesen?

Jeden Monat neu, bei Ihrem PD-Anbieter. (Versand: 1,40 DM Porto)

PD-Pool sucht noch einige gute Programme zur Veröffentlichung auf den Disketten 2051 - 2060. Die Vorstellung erfolgt gleichzeitig in mehreren großen ST- und PD-Zeitschriften.

Die vorgestellten Disketten erhalten Sie exklusiv bei folgenden PD-Anbietern:

HD Computertechnik
Pankstr. 42
1000 Berlin 65
030/4657028-29

trendDATA
Am Marstall 18-22
3000 Hannover 1
0511-166051

Intersoft
Nohlstr. 76
4200 Oberhausen 1
0208-809014

Computer-Software Markert
Balbachtalstr. 71
6970 Lauda 9
Tel: 09343/3854

=PD-EXPRESS= J. Rangnow
Ittlinger Straße 45
7519 Eppingen-Richen
07262/5131 (ab 17 Uhr)

V.U. - Volker Uecker
Hohenkamp 2
2308 Preetz
Tel: 04342-83842

H&S Wohlfahrtstätter
Irenenstr. 76c
4000 Düsseldorf 30
Tel: 0211-429876

Schreiber Software
Josefstr. 27
5120 Herzogenrath
02406-3223

Weeske Computer
Potsdamer Ring 10
7150 Backnang
07191/1528-29 od. 60076

Software-Service Duffner
Ritterstr. 6
7833 Endingen a. K.
07642-3875

ST Profi-Partner
Mönkhofer Weg 126
2400 Lübeck
Tel: 0451/505367

OHST-Software
Nelkenstr. 2
4053 Jüchen 2
Tel: 02164/7898

IDL Software
Lagerstraße 11
6100 Darmstadt 13
06151-58912

Gauger Software
Buhlstr. 16a
7505 Ettlingen
Tel: 07243 - 31828

LAUTERBACH Software
Josephsplatz 3
8000 München 40
089-2722377

T.U.M.-ST-Soft
Postfach 1105
2905 Edewecht
Tel: 04405/6809

Elektronikversand Michiels
Leloh 24
4056 Schwalmtal
Tel: 02163/4187

Computer Treff
Nettelbeckstr. 12
6200 Wiesbaden
Tel: 06121 - 404302

iks
Schönblickstr. 7
7516 Karlsbad 4
ab 18.00 Uhr 07202/6793

Graf & Schick EDV
Hauptstraße 32a
8542 Roth
Tel: 09171/5058-59

☐ Scheck über DM ____ liegt bei, ich erhalte die Ware verpackungs- und versandkostenfrei.

☐ Per Nachnahme. Nur Inland! (zuzüglich DM 6,- Nachnahmegebühr).

☐ 2001	☐ 2011	☐ 2021	☐ 2031	☐ 2041	☐ 2045
☐ 2002	☐ 2012	☐ 2022	☐ 2032	☐ 2042	☐ 2046
☐ 2003	☐ 2013	☐ 2023	☐ 2033	☐ 2043	☐ 2047
☐ 2004	☐ 2014	☐ 2024	☐ 2034	☐ 2044	☐ 2048
☐ 2005	☐ 2015	☐ 2025	☐ 2035		
☐ 2006	☐ 2016	☐ 2026	☐ 2036		
☐ 2007	☐ 2017	☐ 2027	☐ 2037		
☐ 2008	☐ 2018	☐ 2028	☐ 2038		
☐ 2009	☐ 2019	☐ 2029	☐ 2039		
☐ 2010	☐ 2020	☐ 2030	☐ 2040		

Weitere PD-Disks bitte hier mit Nummer angeben ☐ ☐ ☐ ☐

TeX Paketangebot: Disketten 2049_A, 2049_B plus Treiberpaket 2050_C, D, E, F oder G.

☐ TeX + C	16,- DM
☐ TeX + D	16,- DM
☐ TeX + E	20,- DM
☐ TeX + F	20,- DM
☐ TeX + G	20,- DM

Diskpreis: 8,- DM

unverbindlich empfohlener Verkaufspreis

Lieferung an meine Adresse:

____

____

____

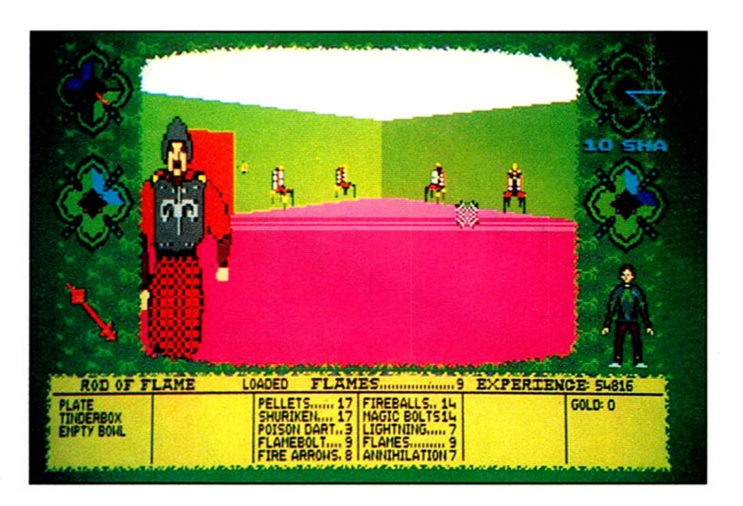
____

STC 10/89

Gewünschte Disketten ankreuzen und Bestellschein an einen der oben angegebenen Anbieter einsenden.

Schnarchender Gott

Tod und Terror sucht das schöne Land Tessera heim: Ein rücksichtsloser Tyrann foltert alle Regimegegner grauenvoll. Die einzige Hoffnung für das geknechtete Volk liegt beim Gott N'Gnir. Nur er kann mit dem raffgierigen Herrscher fertig werden. Dummerweise schläft dieser Gott irgendwo im großen Lande. Es hilft alles nichts! Da muß ein Abenteurer her, den schnarchenden Gott finden und ihn aufwekken. Um an seine Schlafstätte zu gelangen, nimmt der Held eine beschwerliche Reise auf sich. Es müssen acht verschiedene acht verschiedene Königreiche durchquert werden. Jedes ist aus verschiedenen Landschaften zusammengesetzt. Da gibt es Wälder, Gebirge, Seelandschaften und sogar Katakomben. Überall lauern die Schergen des Tyrannen, die versuchen, den Helden aufzuhalten. Anfangs wirft man Steine auf die Widersacher. Später stößt man auch auf effektivere Mordinstrumente wie Armbrüste, Bögen und sogar magische Waffen. Eine ausreichende Bewaffnung ist bei *"Sleeping Gods lie"* unerläßlich. Die Grenztore zu den einzelnen Königreichen werden nämlich von zähen Monstern bewacht. Glücklicherweise treiben sich in



der Abenteuerwelt nicht nur Schurken herum. Es gibt auch friedliche Bewohner in Tessera, die dem Abenteurer helfen, den schlafenden Gott zu finden. Jede Person und alle Landschaften in "Sleeping Gods lie" präsentieren sich in einem großen Fenster, das die Geschehnisse aus der Sicht des Helden darstellt. Ein Fadenkreuz, das sich mit Maus, Joystick oder Tastatur steuern läßt, dient zum Zielen und Abfeuern der Waffen sowie zum Fortbewegen. Dazu bewegt man das Fadenkreuz einfach an den Bildschirmrand, dann scrollt das Bild sauber zum nächsten Teil der Landschaft. Obwohl "Empires" Grafiker durch Bäume, Büsche, Häuser und Seen versucht haben, etwas Abwechslung in den grafischen Aufbau zu bringen, hat man das Gefühl, durch eine Einöde voller trister Farben zu laufen. Aufgelockert wird die langweilige Atmosphäre nur durch die zahlreichen hübsch gezeichneten Charaktere, die man während des Spiels kennenlernt. Von der Grafik einmal abgesehen, macht Sleeping Gods lie einen ausgeschlafenen Eindruck. Viele Personen, Objekte und Interaktionsmöglichkeiten machen diese Mischung aus Adventure, Rollen- und Actionspiel spielenswert. Mich konnte Empires neues Programm zwar nicht vom geregelten Schlaf abhalten, einige unterhaltsame Stunden, habe ich dennoch damit verbracht. Wer nicht vor der tristen Grafik zurückscheut, sollte mal ein Spielchen wagen.

CBO


SLEEPING GODS LIE

Kategorie: Rollenspiel
Besonderheiten: Programmtexte und Anleitung deutsch

Spieler: 1
Monitor: color
Steuerung:
Hersteller: Empire
Info bei: Leisuresoft

Wertung:

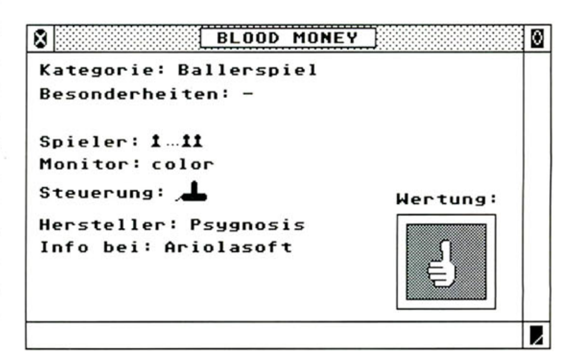
BLOOD MONEY



Blutiges Geld, wer jetzt meint, handle sich um ein äußerst brutales Spiel, irrt. Es ist vielmehr ein weiteres Ballerspiel, das sich im Grunde kaum von alten Klassikern wie *"Katakis"* oder *"Menace"* unterscheidet, letzteres Spiel stammt übrigens von den gleichen Programmieren. In England konnte es große Verkaufszahlen erzielen, in deutschen Landen war der Verkauf jedoch bescheidener. Das Besondere an *"Blood Money"* sind jedoch ein paar extravagante Features. Der oder die Spieler müssen, was sonst, auf alles ballern, was sich vor die Laserkanone wagt. Nach Abschuß verlieren die Widersacher jedoch ihre Ladung, die die Form einer Münze besitzt, sie sollten eingesammelt werden, da das "blutige Geld" zu Extrawaffen verhilft. Dazu müssen Verkaufsplattformen angeflogen werden, die hin und wieder auftauchen. Die Grafik von "Blood Money" ist ansprechend, alle Levels sind gut 'durchgestillt' und durchaus keine Beleidigung für das Auge. Beim Sound haben die Programmierer jedoch etwas gespart. Der Schwierigkeitsgrad von *"Blood Money"* ist recht hoch, am besten spielt man zu zweit gleichzeitig, wobei der eine die Gegner abballert und der andere das Geld einsammelt.

FAZIT: Die Spielidee ist nicht neu, Ballerspielfans kommen aber auf ihre Kosten. Die kleinen Strategieelemente bereichern das Spiel ungemein. Besonders das Spielen zu zweit macht einen Heidenspaß. Vertrieben wird das Ballerspiel von dem bekannten Softwarehaus *"Psygnosis"*, alleine der Anbieter bürgt schon für ein gutes Produkt, stammen doch *"Barbarian"*, *"Obliterator"* oder *"Baal"* aus diesem Hause.

AK


BLOOD MONEY

Kategorie: Ballerspiel
Besonderheiten: -

Spieler: 1...11
Monitor: color
Steuerung:
Hersteller: Psygnosis
Info bei: Ariolasoft

Wertung:

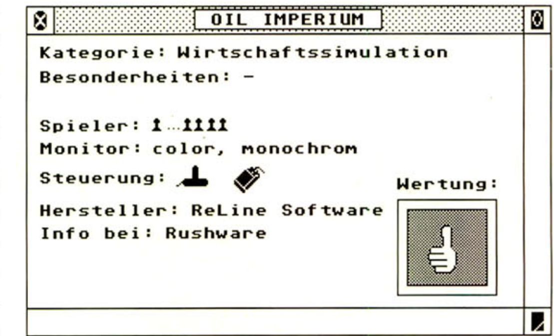
Moneten, Macht und miese Tricks



Normalerweise sind Wirtschaftssimulationen langweilig und trocken. Riesige Zahlenkolonnen, nüchterne Grafik, unzählige Statistiken und komplizierte wirtschaftliche Zusammenhänge machen diese Art von Spiel zwar realistisch, dafür aber langweilig. Ganz anders liegt der Fall bei *"Oil Imperium"*. Reline hat bei diesem neuen Wirtschaftsstrategiespiel weniger Wert auf Realitätsnähe, sondern mehr Sorgfalt auf exzellente Grafik und eine spannende Handlung verwendet. Als Geschäftsführer einer kleinen Ölfirma haben Sie zu Beginn die Wahl zwischen verschiedenen Firmenemblemen und Büroräumen. Geben Sie Ihren Namen ein und los geht der Kampf um die Macht auf dem Ölmarkt. Bis zu vier Spieler können sich an "Oil Imperium" beteiligen. Gewinner ist, je nach Wahl des Spielziels, wer nach drei Jahren über das meiste Kapital verfügt, als erster 60 Millionen Dollar sein Eigen nennt, 80% Marktanteil besitzt oder seine Gegenspieler ruiniert hat. Da wird in jedem ein kleiner J.R. Ewing geweckt. Man kann nämlich Saboteure anheuern und fremde Ölfelder in Brand stecken, die Hausbank der Gegner ausplündern, Handelsvertreter bestechen oder Öltanks in die Luft jagen. Jeder Saboteur kostet Geld. Je billiger sich eine zwielichtige Gestalt auf dem "Markt" anbietet, desto größer ist die Chance, daß er bei Sabotageakten erwischt wird. Das hat unangenehme Folgen. Das Gericht pfändet schuldigen Geschäftsführern, die in schmutzige Geschäfte verwickelt sind, gerne drei Ölfelder. Neben diesen dunklen Machenschaften kann man als Ölmanager natürlich auch neue Felder und Tanks kaufen bzw. verkaufen, Pipelines bauen, nach Öl bohren, sich Informationen über den Ölmarkt aus der Zeitung aneignen, Statistiken anschauen und Lieferverträge abschließen. Gesteuert werden alle Aktionen per Maus mit einer komfortablen Menütechnik. Der ST zeigt das Büro und zwei Leisten mit Symbolen am rechten und linken Bildschirmrand. Möchte man beispielsweise mit dem Schreibtischcomputer geschäftliche Transaktionen durchführen, wählt man das entsprechende Symbol und gelangt auf diese Weise in ein Untermenü, in dem man wieder zwischen verschiedenen Symbolen wählen kann. Texteingaben sind während des Spielverlaufs nicht erforderlich. Alle Vorgänge werden durch wunderschöne Grafiken illustriert. Strategiespielfans haben in noch keinem Spiel dieses Genres eine so gute Grafik zu Gesicht bekommen. Doch "Oil Imperium" ist nicht nur für Strategiefans interessant: Auch an die Fans von Actionsequenzen hat Reline gedacht.

In drei im Programm integrierten Szenen müssen Sie an den Joystick, wenn Sie nicht das Geld für Spezialisten ausgeben wollen. Beim Bohren nach Öl steuern Sie den Bohrkopf und müssen darauf achten, daß er nicht zu stark abdriftet und zerbricht. Ebenfalls mit dem Joystick wird gesteuert, wenn es darum geht, brennende Ölfelder zu löschen. Dabei manövrieren Sie einen Arbeiter, der Bohrtürme sprengen und sich anschließend in Sicherheit bringen muß. In der dritten Szene geht es ums Legen von Pipelines. In diesem Testbericht können wir die mannigfaltigen Optionen von "Oil Imperium" nur streifen. Es gibt noch viele Optionen, die wir aus Platzgründen nicht nennen konnten. Relines neues Strategiespiel überzeugt nicht nur durch gelungene Grafiken, Komplexität und eine spannende Handlung, die einen oder mehrere Spieler stundenlang an den Bildschirm fesselt, sondern auch durch eine komfortable Benutzerführung.

CBO


OIL IMPERIUM

Kategorie: Wirtschaftssimulation
Besonderheiten: -

Spieler: 1...4
Monitor: color, monochrom
Steuerung:
Hersteller: ReLine Software
Info bei: Rushware
Wertung:

Auto-Monitor-Switchbox

Neu: A.R.S. (automatic Resoltution Selection) Mit automatischer Erkennung der Auflösung. Das Starten von Programmen in der falschen Auflösung gehört der Vergangenheit an, Programm anklicken: Programm wird automatisch in der richtigen Auflösung (Farbe oder Monochrom) gestartet.

Softwaremäßiges Umschalten o. Reset erstmals möglich **TASTATURRESET,** und Umschalten über die Tastatur.
Das Umschalten über einen mechanischen Schalter ist selbstverständlich weiterhin möglich.

Auto-Monitor-Switchbox

in neuer Version



Funktion Switchbox

Mit der **AUTO Monitor Switchbox** können Sie über die Tastatur aus jedem Programm heraus zwischen Monochrom und Farbmonitor umschalten oder einen Tastaurreset durchführen. Desweiteren ist es möglich über die Tastatur einen Kaltstart durchzuführen. Die mitgelieferte Software ist resetfest (arbeitet ständig im Hintergrund). Desweiteren ist es möglich durch Einbinden der von uns mitgelieferten Routinen **ohne RESET** zwischen Monochrom und Farbe umschalten.

A.R.S.: Die Software wird nun in einer neuen Version ausgeliefert, die es ermöglicht, daß das Programm automatisch in der richtigen Auflösung gestartet wird. Das ständige Starten von Programmen in der falschen Auflösung gehört der Vergangenheit an.

Die AUTO Monitor-Switchbox verfügt zusätzlich über einen BAS und Audio Ausgang. Die Verarbeitungsqualität wird auch Sie überzeugen.

Update Software: 1.0 auf 2.0 mit A.R.S.
gegen Einsendung der Original Diskette + 15.00 DM.

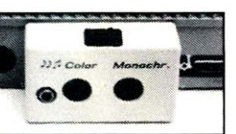
Monitor Switchbox

Monitor Switchbox, die sich nur über einen mechanischen Schalter zwischen Monochrom und Farbe umschalten läßt.

Monitorswitchbox mit 45 cm Kabel, zusätzlicher BAS und Audio Ausgang **44.90**

Monitor Switchbox direkt an den Rechner anstöpselbar mit zusätzlichem Audioausgang **39.90**

dto. als Bausatz (komplett) **29.90**



Switchbox direkt an den Rechner anstöpselbar. (Alle Ausgänge sind weiterhin frei erreichbar. (s. Abb.) Jedoch nur für die Modelle 260/520/1040 ST ohne eingebauten Modulator geeignet.

Preise

Auto Monitor Switchbox **59.90**
mit zusätzlichen BAS u. Audio Ausgang 45 cm Monitorkabel incl. Software

Auto Monitor Switchbox ST **54.90**
direkt an den Rechner anstöpselbar mit zusätzlichem Audio Ausgang incl. Software
Wichtig: nur für die Rechnertypen 260/520/1040 ST ohne eingebauten HF Modulator geeignet.

Auto Monitor Switchbox Multisync **69.90**
incl. Verbindungskabel Multisyncmonitor zusätzlicher Audio Ausgang, incl. Software
Wichtig: Monitortyp angeben

Audiokabel Monitor-Switchbox
2 m 5polig DIN **7.90** 2 m Chinch **9.90**

HF/Videomodulatoren

Video Interface + (in bisher unerreichter Qualität).
Es ermöglicht die Farbwiedergabe des Atari ST an einen Farbfernseher, Monitor oder Videorecorder mit Videoeingang (ideal zum Erzeugen von Videovorspännen). Zusätzlich ist eine AUTO Monitorswitchbox mit einem Monitorausgang integriert (siehe Funktion Auto Monitor Switchbox). Das Gehäuse ist abgeschirmt und atarigrau, Netzteil mit VDE, GS etc. **159,-**

HF Modulator HF Modulator zum Anschluß des Atari ST an jeden gewöhnlichen Farbfernseher. Der HF Modulator zeichnet sich durch die besonders gute Bildwiedergabe aus. Der Ton wird über den Fernseher übertragen.
incl. FBAS, AUDIO Ausgang, Antennenkabel und Netzteil mit VDE, GS etc. **189.-**
Aufpreis Monitorswitchbox 30,00 DM

SIEHE AUCH TESTBERICHT 68000 ST MAGAZIN AUSGABE 1/89

Diskettenlaufwerke

3,5-Zoll und 5,25-Zoll-Diskettenlaufwerke in vollendeter Qualität. Es werden nur die besten Materialien verwendet, z. B. Netzteil vom größten europäischen Netzteilhersteller mit VDE, GS, Thermosicherung. Die Laufwerksgehäuse sind mit einer kratzfesten Speziallackierung ausgestattet. Die Chassis werden **nicht** über den Direktimport bezogen. Dieses ermöglicht eine ständige Lieferbereitschaft und einen guten Service.
Neu: 3,5" wahlweise auch mit den TEAC FD 235 lieferbar. Auf alle TEAC Laufwerke 1 Jahr Garantie. **Qualität und Service, der auch Sie überzeugen wird.**

5,25 Disketten-Laufwerk



Anschlußfertiges 5,25 Zoll Diskettenlaufwerk (720 KB)
Laufwerkstyp **TEAC FD 55 FR,** robuste Verarbeitung. Unterstützt PC DITTO,
Aufpreis 2. Floppybuchse 27.90,
Schaltung A/B 20.-
339.-

3,5 Disketten-Laufwerk



Anschlußfertiges 3,5-Zoll-Diskettenlaufwerk (720 KB)
Laufwerkstyp: **NEC FD 1037**
(25,4 mm hoch) robuste Verarbeitung, kratzfestes Gehäuse Netzteil VDE,
Einbau 2. Floppbuchse 27.90
249.-

Keyboard Interface

Programming PC Keyboard Interface
(Programmierbares PC Tastatur Interface)

Beneiden Sie auch die Tastaturen der großen Computeranlagen, die jeden Tastendruck mit einem unüberhörbaren Klick quittieren? Ziel unserer Entwicklung war ein äußerst flexibles Tastaturinterface, das nicht nur den Anschluß einer PC-Tastatur ermöglicht, sondern das Arbeiten mit den verschiedenen Programmen erleichtert.

So können Sie z. B. beliebig viele Tastaturkombinationen zu einem Makro zusammenfassen und jede Taste der PC-Tastatur damit belegen. Durch ein Start-Up-File, das selbst erzeugt wird, werden die programmierten Makros beim Programmstart automatisch übernommen (das Laden eines zusätzlichen Programmes entfällt). Das heißt: für jedes Programm können Sie Ihre eigenen Makros zusammenstellen.

Leistungsdaten Interface

- Anschluß einer beliebigen PC (XT) Tastatur am ST
- umschaltbare Mehrfachbelegung der PC-Tastatur
- freie Programmierbarkeit der Tastaturbelegungen
- freie Programmierbarkeit von Makros und Generieren von Start-Up Files (mit AUTO Load)
- frei definierbarer Tastaturreset
- unterstützt auch PC Ditto und Rom Port Expander

149.-

Set: Programming PC Keyboard Interface + hervorragende PC-Tastatur mit Mikroschaltern **329.-**

Tastaturabdeckgehäuse
Das Tastaturabdeckgehäuse wird einfach über den Atari ST gestülpt. Alle Ausgänge des Atari sind frei erreichbar. Durch diesen Zusatz wird der Computer zum idealen Monitorständer **59.90**

Hardware Zubehör

Abgesetzte Tastatur am ST
Tastaturgehäuse mit Spiralkabel, Treiberstufe, Resettaste und Joystickportbuchsen eingebaut **124.-**

Leerkarte Speichererweiterung
komplett bestückte Speichererweiterung (steckbar) ohne RAM's
auf 1 MB **84.90** auf 4 MB **249.-**

Speichererweiterung
1 MB bestückt, wieder liefarber (auch für 520 STFM geeignet) **298.-**

Neu: Leerkarte Speichererweiterung auf 2,5 MB **149.-**

Neu im Lieferprogramm: ROM Port Expander und ROM Port Buffer

Software a la carte

Signum II 357.- **STAD** 159.-
PC DITTO Version 3.96 **169.-**
NEU: The best of PD (PD-Paket mit den besten Spielen, Mal, Sound und Anwenderprogrammen auf 5 doppelseitigen Disketten **44.90**



incl. Diskette eins. format.
DM 5,90
ab Nr. PD 225
8,00 DM

PD Pakete 1 Paket = 5 Disk (doppelseitig) **35.-**

- **P.1 Die besten Spielprogramme**
- **P.2 Die besten Utilitieprogramme**
- **P.3 Die besten Mal/Soundprogramme**
- **P.4 Die besten diversen Programme**

Hardware Zubehör

Uhrmodul intern gegenüber anderen Uhrmodulen benötigen Sie keine Software zum Anerkennen der Uhr. Die Bootsoftware befindet sich auf ROM's im Betriebssystem.
Wichtig: Betriebssystem angeben.
ROM TOS o. Blitter TOS **119.-**

Siehe Testbericht 68000 ST Magazin 1/89

Uhrmodul extern incl. Treibersoftware **89.-**

Mouse Pad rutschfeste Unterlage für Computermäuse (270 x 220 mm) **17.90**

3 Laufwerke am ST

Floppyswitchbox: ermöglicht den Anschluß von drei Laufwerken am ST. Das Umschalten erfolgt ohne Reset. Die Switchbox ist mit speziellen Treibern für 3,5 u. 5,25 Laufwerke ausgestattet (m. Zugriffs-LED-Anzeige)
Wichtig: Computertyp angeben
s. Test 68000 ST Magazin 12/88 **89.-**

Multisync Monitore

TVM: die neue Generation von Multisyncs
Graustufen Multisync Monitor (TVM) — die einzige Alternative zum SM 124. Auflösung 900 x 700, alle drei Auflösungen des Atari in SW darstellbar, bestechend scharfe Bildqualität, mit integriertem Schwenkfuß **559.-**

TVM Farbmultisync Monitor 15" Flat Screen
Farbmultisync mit überragenden Leistungsdaten. Alle drei Auflösungen vom Atari darstellbar (Monochrom und Farbe). Das Bild ist um ca. 25 % größer. Sehr gute Wiedergabe auch im Monochrommodus. Der ideale Multisync für den Atari ST
s. Test 6800 ST Magazin 8/89 **1598.-**

Verbindungskabel

Monitorverläng. (1,5 m)	29.90
Harddiskkabel (1,2 m)	39.90
Druckerkabel (2 m)	24.90
Scartkabel (2 m)	39.90
NEU: Romportbuchse	27.90
NEU: Tastaturkabel Mega ST 1,2 m	39.90

Stecker

Monitorstecker	6.40
Monitorbuchse	8.90
Monitorkupplung	9.90
Floppystecker	7.90
Floppybuchse	9.90
Floppykupplung	9.90

Disketten

3,5"-Disketten 2 DD
Qualitätsdisketten (Bulkware) eines namhaften Herstellers
ab 50 Stück **19.90**

Zubehör/Disketten-LW

Floppykabel Atari 3,5 Disketten-LW **27.90**
Floppykabel Atari 5,25 Disketten-LW **29.90**
Floppygehäuse NEC FD 1036, 1037 Teac FD 55 FR, mit Zugentlastung u. Gummifüße **29.-**
NEC FD 1037 mit Anschlußbelegung **197.-**
Teac FD 55 FR **229.-** Atari modif. **239.-**

Hard & Soft A. Herberg, Bahnhofstr. 289, 4620 Castrop-Rauxel, Tel. 0 23 05/1 57 64, Fax. 12022 — Händleranfragen erwünscht —

Soeben eingetroffen: Hardwaremäßiger MS-DOS-EMULATOR (PC-Speed) für den ATARI ST 549.-
Norton-Faktor 4, CGA u. Herkules Emulation (siehe Test 68000 ST Magazin 7/89, c't 89). — 5,25"-Laufwerk + PC Speed 849,- DM — Haben Sie dafür Verständnis, daß wegen der großen Nachfrage zusätzliches Informationsmaterial nur gegen Einsendung eines an sich selbst adressierten DIN-A-5-Umschlags und 2,- DM in Briefmarken zugesendet werden können.



Ein Wort in eigener Sache

In den Jahren, die unsere Zeitschrift existiert, haben wir immer wieder versucht, durch die Beantwortung der bei uns eingehenden Briefe ein wenig Licht in das Dunkel zu bringen, das bei der Arbeit mit dem ATARI ST schon so manch einen aus der Fassung bringen konnte - eine Tatsache, die nicht nur Ihnen, verehrter Leser, sondern auch uns oft genug zu schaffen machte. Nichtsdestotrotz haben wir uns bemüht, die Probleme zu lösen und diverse Leserbriefe zu veröffentlichen, da wir der Meinung waren, daß die jeweilige Thematik auch einen größeren Leserkreis interessieren könnte. Trotzdem gibt es immer wieder Briefe, die wir nicht beantworten können oder dürfen. Damit Sie nicht allzusehr enttäuscht zu sein brauchen oder keine Antwort erhalten, möchten wir Sie bitten, sich an folgende Spielregeln zu halten, die sich aus unserer Erfahrung ergeben haben. Fällt Ihr Brief nicht unter die folgenden Kriterien, hat er gute Chancen, positiv beantwortet oder wenigstens als Hilferuf an unsere Leserschaft gedruckt zu werden.

1. Leider gehen immer wieder Briefe mit dem Wunsch ein, ein Produkt für diesen oder jenen Anwendungsfall vorzuschlagen, verschiedene Produkte bezüglich der Vor- und Nachteile gegeneinander abzuwägen und zu bewerten. Es ist uns aus Wettbewerbsgründen nicht erlaubt, ein bestimmtes Produkt zu favorisieren, selbst wenn wir das eine oder andere in der Redaktion überzeugt einsetzen. Wir können Sie in diesem Fall ausschließlich auf die von uns möglichst objektiven Tests und eventuell anstehende Fachmessen hinweisen. Bedenken Sie bitte, daß auch wir nicht jede Textverarbeitung, jedes Malprogramm und so weiter kennen und bestimmte Produkte dadurch in das Abseits drängen würden.

2. Oft erreichen uns Briefe, die sich positiv oder auch negativ über bestimmte Händler, Softwarehäuser oder deren Produkte auslassen. Sicherlich interessieren uns solche Bemerkungen. Bitte haben Sie aber Verständnis, daß wir weder Lob noch Tadel abdrucken dürfen, da diese Aussagen meist subjektiv sind. Anders sieht die Sache beispielsweise bei Gerichtsurteilen aus, die Sie, verehrte(r) Leser(in), erfochten haben.

3. Aufgrund der Vielzahl an Briefen, die uns täglich erreichen, sind wir leider nicht in der Lage, Programmfehler anhand von Listings oder ähnlichem zu korrigieren. Dennoch sollte ein Problem möglichst detailliert beschrieben sein, denn Ferndiagnosen sind prinzipiell sehr schwer, jedoch mit genauerer Angabe der Symptome eventuell durchführbar.

4. Von Zeit zu Zeit erreichen uns Briefe mit der Bitte, die Adresse des Lesers zwecks allgemeiner Kontaktaufnahme zu veröffentlichen. Würden wir dies in die Tat umsetzen, würde sich der Umfang des anderen redaktionellen Teils beträchtlich verkleinern. Ausnahmen stellen Leser in fernen Ländern dar, für die eine Kontaktaufnahme im eigenen Land recht schierig ist.

Zum Schluß sollen ein paar Tips eventuell voreilig geschriebene Briefe verhindern.

1. Wenn Sie ein Problem bezüglich einer bestimmten Problematik haben oder an einem bestimmten Produkt interessiert sind, finden Sie interessante Artikel darüber eventuell in vorhergehenden Ausgaben userer Zeitschrift. Zur Auswahl eignet sich das Jahresinhaltsverzeichnis besonders gut, das immer am Jahresende in der ST Computer abgedruckt wird.

2. Sollten die Probleme mit der Handhabung eines Produktes zu tun haben, wenden Sie sich zunächst an Ihren Händler und über diesen an den Distributor beziehungsweise an das Software-Haus. Die Wahrscheinlichkeit, daß Ihnen das Software-Haus weiterhelfen kann, ist um ein Vielfaches höher als die, daß wir Ihnen helfen können.

3. Lesen Sie aufmerksam die Leserbrief-Seite. Viele Fragen wiederholen sich immer wieder, obwohl wir bestimmte Probleme schon mehrfach angesprochen haben.

Mal wieder 1 1/2 zeilig in 1st Word Plus

Da jeden Monat einige Briefe bezüglich des eineinhalbzeiligen Schreibens mit 1st_Word Plus eingehen, soll an dieser Stelle exemplarisch ein Leserbrief abgedruckt werden, in dem eine Lösung dargestellt ist. Trotzdem möchte ich darauf hinweisen, daß wir es schon mehrfach an dieser Stelle veröffentlicht haben. Vielleicht hat ATARI irgendwann ein Einsehen und legt ein Beiblatt dieser Textverarbeitung hinzu, in der das Schreiben mit einem Abstand von eineinhalb Zeilen beschrieben wird.

Leserbrief:
Durch Zufall habe ich ein altes ST-Computer-Heft, Mai 88, in die Hände bekommen, in dem ein Leser ausgezeichnet die Vorzüge und Nachteile von 1st Word (Plus) bespricht.
Für eine seiner Forderungen habe ich eine ausgesprochen gangbare Lösung gefunden, nämlich den Ausdruck in anderthalbzeiligen Abständen. Vielleicht wären noch andere ST-Leser an dieser Lösung interessiert.
Ich habe dazu die helle Schrift, die auf dem Drucker ohnehin nicht gut kommt, zweckentfremdet. Vom Installieren des Druckers her kennen wohl die meisten die Drucker-'HEX'-Datei und haben sicherlich eine für ihren Drucker auf der Diskette. Diese lädt man sich ein und ändert unter der Überschrift die Zeilen für E und F. Für meinen NX-1000 (entspricht dem ST LC10) sieht das dann folgendermaßen aus:

```
E, 1B, 32        * Zeilenabstand
                   1/6 Zoll (F4 an)
F, 1B, 33, 36    * Zeilenabstand
                   1/4 Zoll (F4 aus)
```

Da aber nicht alle Drucker die gleichen Befehlssequenzen haben, muß dann wohl jeder sein eigenes Druckerhandbuch zu Rate ziehen. Danach stellt man mit Install.prg eine Druckerkonfiguration her, die man entweder mit einer Kopie von 1st_Word auf einer extra Diskette führt, oder es besteht die Möglichkeit, die Konfiguration je nach Bedarf mit Hilfe der Desktop-Menüleiste umzubenennen (die gerade gültige Variante heißt '.CFG', die andere nach Wunsch).

Bei gleichmäßigem Gebrauch(!) der 1.5zeiligen Abstände zeigt meine Erfahrung, daß man mit einer Seitenlänge von 47 gut über die Runden kommt.

Charlotte Brückner, CH-4133 Pratteln

*

GDAT-Line-A-Anmerkung

Es hat mich sehr gefreut, einen ausführlichen Test unserer schnellen Line-A-Grafik unter APL in Ihrer Zeitschrift lesen zu dürfen. Einige Details bedürfen jedoch der Richtigstellung.

1. Die Programmiersprache APL wird zwar von GDAT vertrieben, Hersteller ist jedoch MicroAPL LTD., London. Wir möchten uns da nicht mit fremden Lorbeeren schmücken.

2. Zum Ausdrucken eines Bildschirmauschnittes, wie im Beispiel (8) beschrieben, ist es nicht nötig, eine Variable zu initialisieren. Die erste Zeile des Beispiels kann also ersatzlos entfallen.

3. Die Textblt-Routine arbeitet mit den drei Systemzeichensätzen des ATARI, nicht mit dem APL-Zeichensatz. Sie unterscheidet sich von der ursprüglichen Line-A-Routine nur dadurch, daß sie auch Textstrukturen (Vektoren und Matrizen) verarbeiten kann.

4. Es stimmt, daß bei der Benutzung der Line-A die sichere APL-Umgebung verlassen wird. Dies liegt aber in der Natur der Line-A, die auf der untersten Ebene der Betriebssystemgrafik angesiedelt ist.

5. Einzelne Line-A-Aufrufe können tatsächlich nicht unterbrochen werden. Dies ist jedoch auch recht unsinnig, da selbst komplexe Operationen wie das Zeichnen von über tausend Vektoren nur Sekundenbruchteile in Anspruch nehmen. Dies ist eine Zeitspanne, die kaum ausreicht, die Finger auf die passenden Tasten zu bekommen. Zwischen den Aufrufen einzelner Line-A-Funktionen ist eine Unterbrechung jederzeit möglich.

Ich bitte darum, diese Punkte nicht als Mäkelei am Artikel mißzuver-

stehen. Der Autor hat sich sehr eingehend mit der Materie beschäftigt und die wesentlichen Eigenschaften unserer Line-A-Bibliothek korrekt und einfühlsam beschrieben. Weiter so!

M. Burghardt, Geschäftsführer der gdat mbH, Bielefeld

*

Preiswerte Fremdtastaturen und 1 Megabyte im STFM

Ihre Antwort auf den Leserbrief von Herrn Gert Korn in Heft 4/89 hat mich doch etwas verwirrt, da Sie schreiben, zu vergleichbaren Preisen wie denen der Tastaturkappen der Fa. RTS-Elektronik seien schon AT-Tastaturen zu bekommen. Dagegen nennen Sie in Ihrem Bericht in Heft 12/88 bezüglich Fremdtastaturen am ST Preise von ca. DM 250,- und aufwärts. Gibt es also noch günstigere Fremdtastaturen (für mich als vielschreibenden Studenten natürlich interessant), die entsprechend kompatibel sind?

Eine andere Frage: Ich möchte meinen 520STFM auf ein MEGA aufrüsten. Könnten dabei Probleme mit Soft- und Hardware auftreten?

Mathias Klodt, 2408 Niendorf

Red.: Zunächst mal muß natürlich eins gesagt werden: Die Leserbriefe werden meist nicht von den Mitarbeitern der Redaktion beantwortet, die auch die dazu passenden Artikel schreiben, da dies organisatorisch kaum machbar ist. Zu der Aussage, daß es günstige AT-Tastaturen unter 250,-DM gibt, stehe ich trotzdem, auch wenn ich Ihnen aus Wettbewerbsgründen keinen Händler nennen darf. Ein Tip: Schauen Sie einmal in unseren Anzeigenteil, oder, falls Sie dort nichts finden, nehmen Sie sich am besten einmal eine oder zwei MS-DOS- oder Elekronik-(Computer)-Zeitschriften vor. Beachten Sie aber, daß Sie diese Tastaturen nicht direkt an den ST anschließen können, sondern daß noch die entsprechende Hardware nötig ist, um diese Tastatur anzupassen.

*

Feste Returns anstatt Spaces in Wordplus

In Ihrer Antwort auf den Leserbrief 'Feste Spaces in Wordplus' von Herrn Trolldenier in der ST-Computer 5/89 geben Sie ein verblüffend einfaches Rezept zur Neuformatierung von ASCII-Texten in Wordplus an.

Ich habe es ausprobiert mit Texten, die ich mit Signum2 erstellt und als ASCII-Files abgespeichert hatte. Es funktioniert tatsächlich: Wenn man 'Space durch Space' ersetzt, macht Wordplus variable Textzwischenräume daraus. Trotzdem ist nur die Hälfte des Problems gelöst, denn leider betrachtet Wordplus jede Zeile des ASCII-Textes als separaten Absatz und macht daher eben keinen flüssigen Zeilenumbruch. Der Versuch, 'Return' durch 'Return' zu ersetzen, ist von vorneherein zum Scheitern verurteilt, weil man mit 'Return' sofort aus dem Menüpunkt 'Ersetzen' herausfliegt. Können Sie auch hier eine so schön einfache Lösung vorschlagen?

Dr. Dietrich Panke, Röhm Gmbh, 6100 Darmstadt

Red.: Ja! Die Lösung ist ganz einfach: Auf der PD-Disk 78 befindet sich ein Programm, das CON_WP heißt und in Nullkommanichts Ihren Text umsetzt.

*

Kopierschutzfrage - BERICHTIGUNG

Bei der Durchsicht der ST-Computer Mai 1989 bin ich auf den Leserbrief von Herrn Wietschorke aus Karlsruhe und Ihre Anwort hierzu gestoßen. Da ich als Jurist, der gerade auf dem Gebiet "Kopierschutz von Computerprogrammen" promoviert, mit diesem Thema zur Zeit intensiv befaßt bin, möchte ich zu Ihrer Antwort Stellung nehmen. Meines Erachtens ist die von Ihnen gegebene Auskunft teilweise fehlerhaft und könnte bei Rechtsunkundigen zu Fehleinschätzungen führen, die ihrerseits juristische Konsequenzen auslösen können.

Zutreffend sind die Ausführungen insoweit, als Sie darauf hinweisen, daß teilweise Verkaufsbedingungen zugrundegelegt werden, in denen sich die Anwendung des gekauften Programms auf den lizensierten Erwerber beschränkt. Dies ist zum Beispiel gerade bei SIGNUM!2 der Fall, bei dem das Softwarehaus eine Anwendung nur durch den Käufer erlaubt, so daß das von Herrn Wietschorke gewünschte und von Ihnen begrüßte Verfahren nicht möglich ist.

Falsch jedoch ist Ihre Aussage, Software könne wie ein Buch behandelt werden, jedenfalls dann, wenn es sich nicht um PD-Programme handelt, auf die das Urheberrecht nicht anwendbar ist. Im folgenden möchte ich versuchen, die Vorschriften des Urheberrechts (UrhG) allgemeinverständlich zu erläutern, soweit sie für die aufgeworfene Frage von Belang sind. Jeder Urheber wird durch das UrhG insofern geschützt, als ihm die Verwertungsrechte nach §§ 15 bis 24 UrhG zustehen. Als wichtigstes Recht im Bereich der Software ist hier das Vervielfältigungsrecht nach § 16 UrhG zu nennen. Vervielfältigung in diesem Sinne ist nicht nur das Erstellen einer Sicherungskopie, sondern schon das Laden des Programmes in das RAM des Rechners. Hieraus folgt, daß eine "buchartige" Benutzung von Software nur dann zulässig ist, wenn die Vielzahl von Benutzern hierzu befugt ist. Nach dem System des UrhG ist dies aber nur der Käufer. Diesem wird nämlich durch den Kauf gemäß § 31 Abs. 1 UrhG das Nutzungsrecht eingeräumt, und zwar in der Regel (bei Standardsoftware) das einfache Nutzungsrecht nach § 31 Abs. 2 UrhG. Dieses Nutzungsrecht erlaubt dem Käufer, das Programm neben dem Urheberrechtsinhaber und anderen Nutzungsberechtigten im vertraglich geregelten Rahmen zu nutzen. Ein Recht der Weitergabe kann hieraus jedoch nicht abgeleitet werden! § 34 Abs. 1 UrhG schreibt vor, daß das Nutzungsrecht nur mit Zustimmung des Urhebers weitergegeben werden kann und der Verkäufer diese Zustimmung nicht wider Treu und Glauben verweigern darf. Ohne konkret auf die Frage von Treu und Glauben eingehen zu wollen, kann man das wirtschaftliche Interesse des Urhebers nicht als Verstoß gegen Treu und Glauben ansehen. Hieraus folgt, daß die von Ihnen erwähnte "buchartige" Nutzung immer dann ein Verstoß gegen das UrhG darstellt, wenn dem Käufer die Berechtigung zur Weitergabe nicht ausdrücklich vertraglich eingeräumt wurde. Dies ist aber meines Wissens bei Standardsoftware regelmäßig nicht der Fall.

Es darf hier auch nicht übersehen werden, daß ein Buch, welches weitergegeben wird, keine Vervielfältigung erfährt. Da aber ein Computerprogramm nach § 16 UrhG bereits dann vervielfältigt wird, wenn es in den Arbeitsspeicher eingelesen wird, verbietet sich im Hinblick auf das UrhG eine Gleichbehandlung von Büchern und Software.

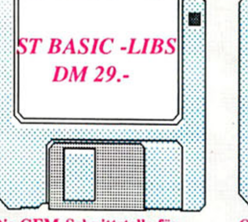
Wegen der Strafvorschrift der § 106 UrhG, der eine Höchststrafe von einem Jahr Freiheitsstrafe für die unerlaubte Weitergabe und Benutzung urheberrechtlich geschützter Werke vorsieht und somit sowohl Käufer als auch unberechtigten Benutzer bedroht, möchte ich Sie bitten, Ihrer Stellung eine Korrektur anzufügen, zumal Ihr Magazin auch von Jugendlichen gelesen wird, die sich teilweise über die Rechtsfolgen ihres Handelns noch nicht bewußt sind und bei falscher Auslegung ihrer Antwort sich unbewußt einer Gefahr aussetzen würden. Die zivilrechtlichen Folgen dürften ebenfalls nicht unbeachtlich sein.

R.-Ref. Derk U. Becker, Ludwigshafen

Red.: Wir bedanken uns hiermit sehr herzlich für die fachkundige Rechtsbelehrung in diesem auch in der Rechtswelt noch sehr neuen Gebiet und kommen gerne dem Wunsch der Veröffentlichung nach. Vielleicht bemerken dabei auch diejenigen Jugendlichen, die völlig bewußt(!) raubkopieren, welche Folgen dies für sie haben kann. Die Tatsache, daß es 'heute jeder' macht, legalisiert es nicht, sondern verschleiert höchstens eine Straftat. Die oben angesprochene Freiheitsstrafe von einem Jahr sollte eigentlich genug Abschreckung sein!

*

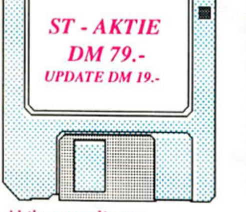
STARKE SOFTWARE


ST BASIC -LIBS
DM 29.-
Die GEM-Schnittstelle für GFA-Basic 2.0


TKC-HAUSHALT ST EXPERT
DM 129.-
Ordnung, Überblick, Planung mit der Haushaltsbuchführung für alle

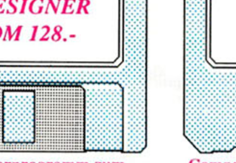
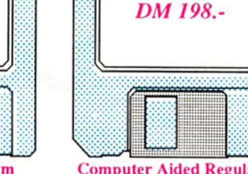

TKC-EINNAHME / ÜBERSCHUSS ST-EXPERT
DM 149.-
Die komfortable Buchführung für Freiberufler und Kleingewerbetreibende

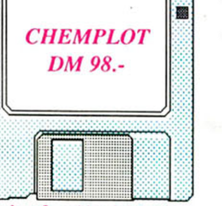

ST PRINT
DM 69.-
UPDATE DM 19.-
Das vielfach bewährte Multi-accessory jetzt in erweiterter Version V2.1 incl. Viruskiller

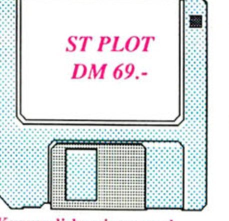

ST - AKTIE
DM 79.-
UPDATE DM 19.-
Aktienverwaltung
Neue Version 2.6

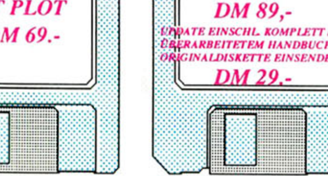
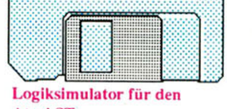

ST - MAXIDAT
DM 98.-
Die Dateiverwaltung für den hohen Anspruch

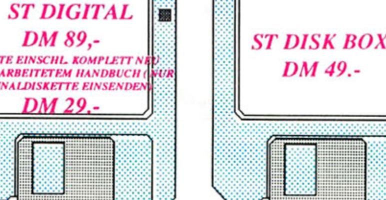

ST KREATIV - DESIGNER
DM 128.-
Das Superprogramm zum Kreativen erstellen von Grafiken aller Art
- ohne Malkenntnisse



ST C.A.R.
DM 198.-
Computer Aided Regulation


CHEMPLOT
DM 98.-
Ein komfortables, leicht zu bedienendes Zeichenprogramm für chemische Moleküle


ST PLOT
DM 69.-
Kurvendiskussions - und Funktionsplotter Programm



ST DIGITAL
DM 89,-
UPDATE EINSCHL. KOMPLETT NEU ÜBERARBEITETEM HANDBUCH (NUR ORIGINALDISKETTE EINSENDEN)
DM 29.-
Logiksimulator für den Atari ST
- Neue Version 2.0 -


ST DISK BOX
DM 49.-
Disketten - Verwaltungsprogramm

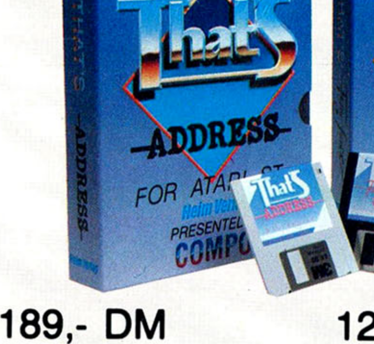
ST ARCHIVAR
DM 89.-
Die überzeugende Dateiverwaltung

LABEL CONSTRUCTION SET
DM 69.-
Gestaltung eigener 3,5" Diskettenlabels

TKC FAKTURA V 1.6
DM 398.-
Der neue Masstab in sachen Geschwindigkeit und Benutzerfreundlichkeit



ST AKTIENSTAR
DM 198.-
Das profisionelle Aktien - Verwaltungsprogramm



189,- DM

That's Address, die Adressverwaltung zu That's Write

- ☐ Direkte Übernahme einer Adresse von That's Address in den Brief ohne die Daten noch einmal schreiben zu müssen, oder das Programm zu verlassen.
- ☐ Aus diesen Adressen nach eigenen Kriterien bestimmte Einträge für einen Serienbrief selektieren.

Unter Berücksichtigung dieser Vorgaben für That's Write haben wir **That's Address** entwickelt:

Ständig verfügbar
That's Address ist ein **Accessory**, d. h. aus einem laufenden Programm kann auf die Daten zugegriffen werden. Das Programm wird beim Einschalten des Rechners einmal automatisch gestartet und bleibt bis zum Ausschalten verfügbar.

oder als Programm
That's Address läuft auch als Programm, statt dem Accessory kann es (bei zu geringem Speicherplatz z. B.) als Programm gestartet werden.

Hohe Datensicherheit
Jede Änderung oder Neuaufnahme einer Adresse wird sofort abgespeichert.

Einfache Bedienung
wahlweise per Maus oder Tastatur lassen sich alle Funktionen betätigen.

Adressübergabe an That's Write
Wählen Sie den Knopf **Übergabe** – Sie wechseln automatisch zurück in die Textverarbeitung und die angezeigte Adresse erscheint an der gewünschten Stelle in Ihrem Text.

Serienbrief/Datenexport
Ausgewählte Adressen (siehe Selektion) können an That's Write übergeben werden. In Verbindung mit einem von Ihnen gestalteten Text macht That's Write daraus ein Rundschreiben.

Selektion
Adressen können nach freien Kriterien ausgewählt werden. Ausgewählte Listen können sofort bearbeitet werden oder zur späteren oder mehrfachen Verwendung (z. B. erst bearbeiten, dann Etikett, Serienbrief) abgespeichert werden.

Manuelle Selektion
Einzelne Adressen können von Hand (auch mehrfach) z. B. für Etikettendruck abgespeichert werden

Formulardruck
Per Knopfdruck bedruckt That's Address Briefumschläge, Karteikarten, Überweisungen mit beliebigem Absender/Empfänger

Listen-/Etikettendruck
Ausgewählte Adressen können auf den Drucker als
- Endlos-Etiketten
- Liste

ausgegeben werden.

Formularanpassung
Die Formulare wie Überweisung,... können über That's Write von jedem selber geändert werden.

Übersichtlicher Bildschirmaufbau
- Alle Feldbeschriftungen wie Straße, PLZ, Ort,... sind in kleiner Schrift gehalten (man kennt die Felder nach mehrmaliger Benutzung sowieso auswendig) und können vom Benutzer einfach geändert werden.
- Alle Daten wie **Peter Müller, Dorfstr. 34, ...** sind in normaler Schrift gehalten.

Dadurch wird der Bildschirmaufbau nicht überladen und wichtige Daten sind auf einen Blick zu erkennen.

Für den Einsatz von That's Address empfehlen wir einen ATARI ST mit 1 MB RAM (läuft ohne That's Write notfalls auch mit 512 kB)

128,- DM

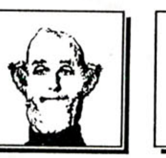
Das Gesichtsmenü 5.576.787.923.200.00 Gesichter

Quick Brown Fox ©1989
if He knew of this before,
the world surely would look a lot more fun












Presented by COMPO

Läuft auf allen ATARI St/Mega in hoher, mittlerer und niedriger Auflösung (auch auf Stacy) ● Schieben Teile links, rechts, auf, ab, nach innen und außen ● Auswahl per Ziffer oder Zufallsgenerator ● Zufallsanimation ● Sichern und Laden als FAC ● GRAFIK MENÜ: Freihand, Sprühdose & Füllen mit Muster ● Primitives ● Text ● Vergrößern ● Ausschneiden, Kopieren & Einfügen von/nach Clipboard ● Outline ● Bas-Relief ● Spiegeln & Flip-Flop ● Abdunkeln & Aufhellen ● Pixelate & Raster ● Stauchen & Strecken ● Schatten & Multischatten ● 5 Bildschirme ● Sichern/Laden von IMG, PIx, RGH, CLP & ICN

328,- DM

Die professionelle Textverarbeitung für Atari ST

Endlich... ein Textverarbeitungsprogramm, das neben seinem Können auch anwenderfreundlich ist

Mit **That's Write** steht Ihnen ein flexibles und umfangreiches Textprogramm für Ihren ST zur Verfügung. **That's Write** bietet sich für alle gängigen Anwendungen einer Textverarbeitung an. Durch die einfache Bedienung per Maus ist es für den Einsteiger leicht und schnell zu erlernen, bietet aber für den vielschreibenden Anwender ebenfalls die Bedienung über die Tastatur oder die Definition eigener Funktionen über Makro.

Änderungen von That's Write 1.3 gegenüber Version 1.2:

1. ***Neues Handbuch*** (über 200 Seiten im Ringordner mit Schuber)
2. ca. ***60kB*** kürzer
3. ca. ***20-30%*** schneller (Bilder umrechnen sogar 2 bis 10 mal schneller)
4. ***OnLine* Rechtschreibkorrektur** Korrektur schon während dem Schreiben)
5. ***Gliederungsfunktion*** (Text kann stufenweise zugeklappt werden)
6. ***Absatzabstand einstellbar*** (Leerzeilen zwischen Absätzen jetzt unnötig)
7. ***Statistikfunktion*** zählt Seiten, Zeilen, Worte, Bilder
8. Bilder einzeln anzeigen, ändern der Größe auch per Maus (Gummiband), **sichern v. Bild auf Disk**
9. Block: sichern nun auch ASCII, geändertes laden/sichern
10. Zeichen - nach Layout schaltet auf Absatzattribute zurück
11. Attribute merken/setzen erlaubt Speicherung der Attribute
12. Stichwort ist nun invers dargestellt, da 'Inhalt' durch Gliederung fast nicht mehr benötigt wird
13. Zählfunktion zählt Worte bzw. Zeichenketten
14. **Seitennummernsymbol für Folgeseite**
15. Absatz-/Seitenlayout übernehmen für neues Layout die Voreinstellung des aktuellen Layouts
16. Ausdruck wahlweise ohne Bilder (Probeausdruck...)
17. Tastenkommandos Wort groß, klein, klein aber 1. Buchstabe groß
18. Wortweise springen nun auch bei silbengetrennten Worten korrekt
19. Absatzweise vor/zurück

Preis: 328,- DM

ST-COMPUTER PUBLIC DOMAIN

Liebe PD-Freunde,

Es ist mal wieder soweit, ein großes Jubiläum steht an. Die 250.te PD-Diskette hat das Licht der Welt erblickt. In mehreren Jahren ist durch die Arbeit vieler Programmierer gelungen, diese fantastische Sammlung zu formen. Es dürfte wohl kaum einen Anwender geben, der nicht ein oder mehrere PD-Programme daraus benutzt. Wir möchten uns im Namen aller PD-User bei den Programmautoren bedanken und hoffen, auch bald die nächsten 250 präsentieren zu können. Dabei muß ich allerdings darauf hinweisen, daß uns nicht an der Menge der Disketten gelegen ist. Zwar freut uns diese Zahl, das Hauptmerkmal liegt aber auf deren Qualität. Würden wir beispielsweise auf Demos, Bildershows oder amerikanische PD-Serien zurückgreifen, wäre diese Zahl sicherlich schon erreicht, aber das haben wir bislang so gut es ging vermieden. Wir denken, das liegt auch in Ihrem Interesse.

Bis zur 500.sten



WellerTools HD

FIND FILE: Programm zum schnellen Suchen einzelner Dateien auf Festplatte oder Diskette unter Berücksichtigung von Wildcards (*,?). Hierbei können bestimmte Partitionen eingegrenzt werden.

FIND TEXT: Suche nach Textbruchstücken innerhalb von Dateien auf einer, bzw. mehrerer Partitionen der Platte oder einer Diskette. Ideal zum Suchen einer Datei, von der man nur den Inhalt kennt (z.B. ein Brief an "Hugo"). Das Programm arbeitet äußerst schnell, sodaß Wartezeiten kaum auftreten.

SETUP: Setup ermöglicht das Auswählen zwischen mehreren Accessories und Autostart-Programmen. Außerdem verlangt es nach Zeit und Datum, falls keine Uhr eingebaut ist.



Libraries Modula/PASCAL

LIBS.LPR: Umfangreiche Bibliothekensammlung für das LPR-Modula-System unserer Sammlung (PD 225). Neben Stringfunktionen, sind erweiterte I/O-Befehle, MathLib, bzw. LongInt-Math und Konvertierroutinen vorhanden.

STRING.LIB: Spezielle String-Bibliothek für das LPR-Modula (PD225).Kopieren, Füllen, Suchen, Einfügen, Länge, ect.

MATHELIB: Umfangreiche Bibliothek für PASCAL aus der Welt der Mathematik und der restlichen Rechnerwelt.

Mathematik: Logarithmen, RAD<->GRAD, trigonometrie, Hyperbolicus, Umkehrfunktionen, Binominalkoeffizient, Komplexe Zahlen, Lösung quadratischer Gleichungen, Polynomgleichungen, Gaussverfahren, Newton, Matrizenrechnungen, Differensation, Integralrechnung, Lösung von Differentialgleichungen, u.a.
Kurvendiskussion: Zeichnen von Koordinatensystemen, Achsenkreuze und gesamte Funktionen, komplette Kurvendiskussionen (Nullstellen, Max., Min., Wendepunkte und Sprünge nach verschiedenen Verfahren sowie die formatierte Ausgabe der Ergebnisse.
Sonstiges: Zahlen in Strings, Formatierte Ausgabe von Vektoren und Matrizen, Soundbefehle, VT52, LINE A, u.a.

Wir sind auch weiterhin an Bibliotheken zu allen Programmiersprachen interessiert. Hierbei kann es sich auch um kleinere Routinen handeln. Einsendungen werden nach Möglichkeit auf themenorientierten Disketten zusammengestellt.

Einsendungen an:

MAXON Computer
PD-Libraries
Postfach
6236 Eschborn



Textutilities

SNAPFONT: Ermöglicht große Überschriften für Signum!. Die erzeugten Überschriften können als Grafik in beliebiger Größe in Signum!-Texte eingebunden werden. Weiterhin wird die Wandlung von Grafik in Signum!-Fonts oder die Verfremdung bestehender Fonts durch Malprogramme unterstützt. Wer möchte, kann sich auch seine Fonts im Malprogramm entwerfen und mit SNAPFONT nach Signum! konvertieren.
Ein Muß für Signum!-Anwender. Mit Source-Code in OMIKRON.BASIC (s/w)



Ein Überschrift mit geladenem Signum!-Font

INHALT_INDEX: Programme zur Erzeugung von Inhalts- und Indexverzeichnissen von Texten. Was ist lästiger, als dies per Hand machen zu müssen? Die Programme nehmen diese Arbeit ab und sind sehr nützlich bei Referaten, Büchern oder sonstigen längeren Texten.

LINGUIX: Komplexe Programmsammlung zur Textbearbeitung. Indexlisten, Zeilennumerierung, Wortlisten, Sortieren, Ersetzen, Suchen und Anzeigen im Kontext, ASCII -> WP, Zeilenindex, Inhaltsverzeichnisse, Vergleicht Dateien, Zählen von Zeichen, Wörtern und Zeilen, Filtern, und vieles mehr.

PHRASER II / LASTWORD: Programme zur Aufwertung von Referaten, Seminararbeiten oder sonstigen Arbeiten. PHRASER beinhaltet ca. 8000 Formulierungen die nach dem Zufallsprinzip zusammengewürfelt werden. Die Ausgabe auf Drucker oder RS 232 ist möglich.
LASTWORD arbeitet nach selbigem Prinzip, fügt die Ausdrücke jedoch direkt in das gerade benutzte Textprogramm ein.

Einige Beispiele:
progressive Aktionsdualismen
kompatible Fluktuationskompetenz
partielle Eruptionssuffizienz
tradierte Falsifikationssynthese
elitäre Exemplifikationseventualität
divergierende Kooperationsstruktur
emanzipatorische Uniformitätsdeterioration

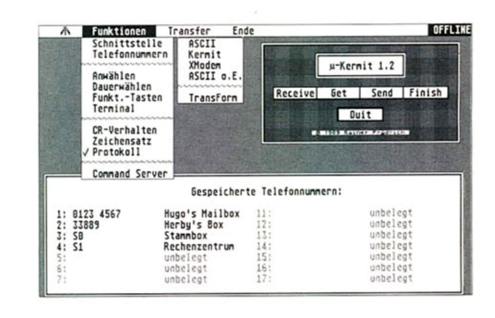
Umhüllt von einem Satz wie etwa:
Im Rahmen der....spielt die....eine bedeutende Rolle, was eine....nach sich zieht.

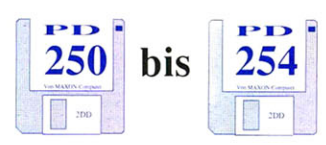
wird jeder Text zunehmend interessanter (im liberal inkohärenten Sinne, versteht sich)



Terminal

TRANSTERM: Mit Transterm erhalten Sie ein sehr nützliches Terminalprogramm. Nicht nur, daß es ein pures Terminalprogramm wäre, nein: Sie können beliebige Dateien mit dem XModem- und sogar dem Kermit-Protokoll überspielen. Alle wichtigen Programme, die Sie zur DFÜ benötigen, sind bei Transterm gleich mit dabei, so z.B. auch das Archiv-Programm ARC in der Version 5.21 inklusive 80k Dokumentationstext, davon allein 30k für Transterm. Transterm ist eine echte Alternative zu Uniterm (PD 117). (s/w)



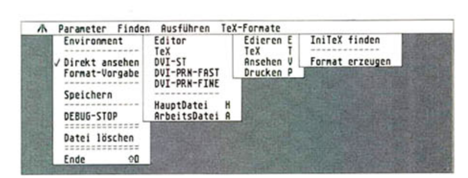


TeX-System

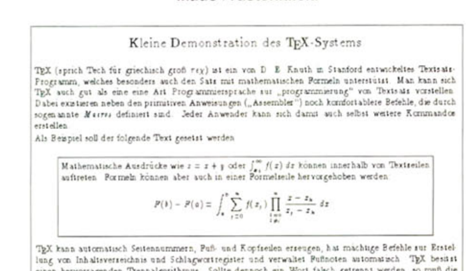
Schon oft haben wir über käufliche Versionen geschrieben, nun freuen wir uns, Ihnen ein komplettes TeX-System zu präsentieren, das Ihren ST in ein professionelles Satzsystem verwandelt. Das System besteht auf fünf randvollen Disketten auf denen das System, viele Zeichensätze, Druckertreiber, Fonteditor und Shells untergebracht sind.
(s/w, mind. 1MB)

250 251

TeX-Grundpaket: Auf den beiden Disketten befindet sich das TeX-Hauptprogramm, PLAIN und LaTeX (komfortable Macros für komplizierte Darstellungen), deutsch und englische Trennungen (sehr leistungsstark), Druckertreiber für NEC P6 (180*180 dpi) und eine komfortable Shell, die das Arbeiten mit TeX, was sonst nur über umständliche Textbefehle funktioniert, regelt.
Für die Ausgabe des Textes auf dem Bildschirm sorgt der beigefügte Bildschirmtreiber, auf dem das fertige Dokument zu sehen ist.



Die TeX-Shell regelt die Steuerung des Systems. Übersetzen, Anzeigen Drucken - alles per Maus-/Tastenklick.



Kleine Demonstration des TEX-Systems

TEX (sprich Tech für griechisch groß $\tau\epsilon\chi$) ist ein von D. E. Knuth in Stanford entwickeltes Textsatz-Programm, welches besonders auch den Satz mit mathematischen Formeln unterstützt. Man kann sich TEX auch gut als eine eine Art Programmiersprache zur „programmierung" von Textsatz vorstellen. Dabei existeren neben den primitiven Anweisungen („Assembler") noch komfortablere Befehle, die durch sogenannte *Macros* definiert sind. Jeder Anwender kann sich damit auch selbst weitere Kommandos erstellen.
Als Beispiel soll der folgende Text gesetzt werden

> Mathematische Ausdrücke wie $z = x + y$ oder $\int_{x_1}^{\infty} f(x)\,dx$ können innerhalb von Textzeilen auftreten. Formeln können aber auch in einer Formelzeile hervorgehoben werden
>
> $$F(b) - F(a) = \int_a^b \sum_{i=0}^{n} f(x_i) \prod_{k=0, k\neq i}^{n} \frac{x - x_k}{x_i - x_k}\,dx$$

TEX kann automatisch Seitennummern, Fuß- und Kopfzeilen erzeugen, hat mächtige Befehle zur Erstellung von Inhaltsverzeichnis und Schlagwortregister und verwaltet Fußnoten automatisch. TEX besitzt einen hervorragenden Trennalgorithmus. Sollte dennoch ein Wort falsch getrennt werden, so muß die korrekte Trennung nur *einmal* im Text eingegeben werden.

Ein Text auf dem Bildschirm dargestellt.

252

PD-TeX-Erweiterung 1: Druckertreiber für NEC P6 in hoher Auflösung (360*360). Auch dieser Treiber wird direkt von der Shell aufgerufen und liefert optimale Druckqualität.

253-254

PD-TeX-MetaFont: MetaFont dient zur Erzeugung und Umrechnung von TeX-Fonts. Dabei können Fonts für beliebige Ausgabegeräte, in verschiedenen Größen und verschiedenen Styles erzeugt werden.

Was ist TeX?

Bei TeX handelt es sich um ein Satzsystem, das sich deutlich von Textverarbeitungen abhebt. Als kleines Beispiel sehen Sie unten eine komplexe Formel, welche von TeX nach Eingabe der mathematischen Formel eigenständig berechnet wird. TeX verwaltet weiterhin Ihre Fußnoten, erstellt Inhaltsverzeichnisse, Seitennumerierungen, und Inhaltsverzeichnisse. Weiterhin verfügt es über automatische und sehr leistungsstarke Trennalgorithmen und vor allem über ein sehr ansprechendes und professionelles Schriftbild, wodurch TeX häufig zum Erstellen von Büchern verwendet wird. Aber auch zum Schreiben von eindrucksvollen Büchern ist TeX zu empfehlen. Ein Artikel über die Grundlagen dieses Systems war in ST-Computer 5'89 zu lesen.

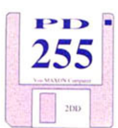
$$F(b) - F(a) = \int_a^b \sum_{j=0}^{n} f(x_j) \prod_{\substack{k=0 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k}\, dx \; .$$

nein — klar ? — ja

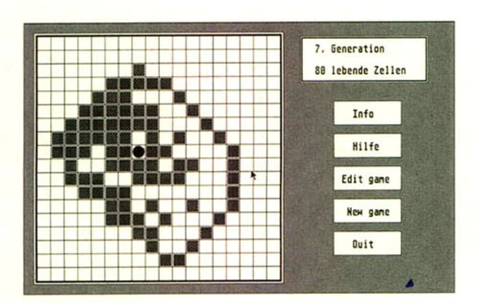
schlecht | gut

$$\lim_{n\to\infty} \underbrace{\sqrt{2+\sqrt{2+\cdots+\sqrt{2+\sqrt{2}}}}}_{n \text{ Wurzeln}} = 2$$

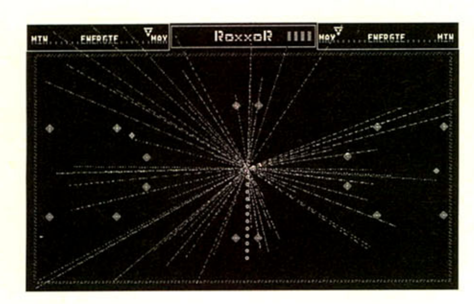
Die Testausdrucke zeigen die Leistungsstärke von TeX



Spiele

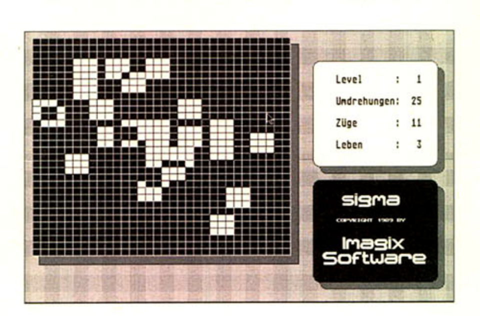


KILLER: Neue Dimensionen für das altbekannte LIFE. Der Spieler versucht, ziehend nach dem Springer-Prinzip, das Ausbreiten von Zellen zu verhindern. Trifft er keine Zelle, so vermehrt sich die Zellenkolonie. Es gilt, gezielt zu springen, sonst geraten die Zellen außer Kontrolle. (s/w)

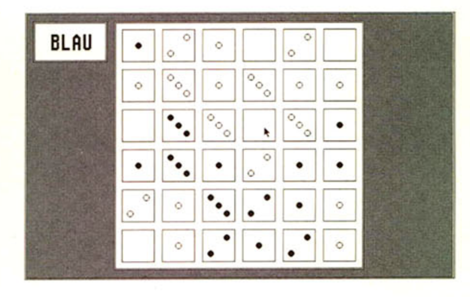


ROXXOR: Stilvolle TRON-Variante. Zwei Spieler treten mit ihren Wagen gegeneinander an, um sich zu zerstören. Das Spielfeld ist mit etlichen Hindernissen geradezu gepflastert, so das das Spiel zu einer Herausforderung wird. (s/w)

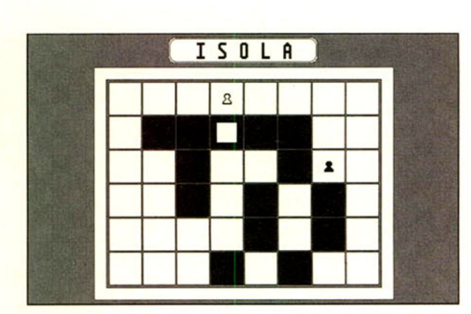
Ab Diskette 225 werden alle ST Computer-PD-Disketten doppelseitig ausgeliefert!



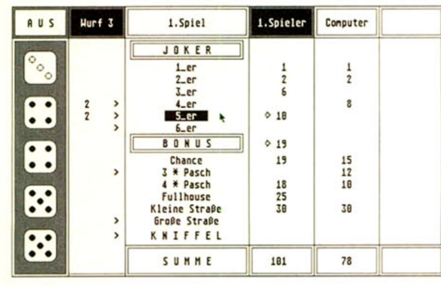
SIGMA: Diffiziles Spiel gegen den Computer, das Sie sich am besten selbst anschauen. Nach einiger Spielzeit ist man jedenfalls davon gefesselt. (s/w)



EXPLODE: Ein Spiel zu zweit. Die Spieler besetzen verschiedene Felder mit ihren Spielsteinen. Hat ein "normales" Feld mehr als drei Steine zu fassen, explodiert es und die dieses Feld umgebenen Spielsteine wechseln den Besitzer.



ISOLA: Jeder Spieler versucht den anderen einzubauen, indem er ihm Hindernisse in den Weg baut. Natürlich kann auch gegen den Computer gespielt werden.

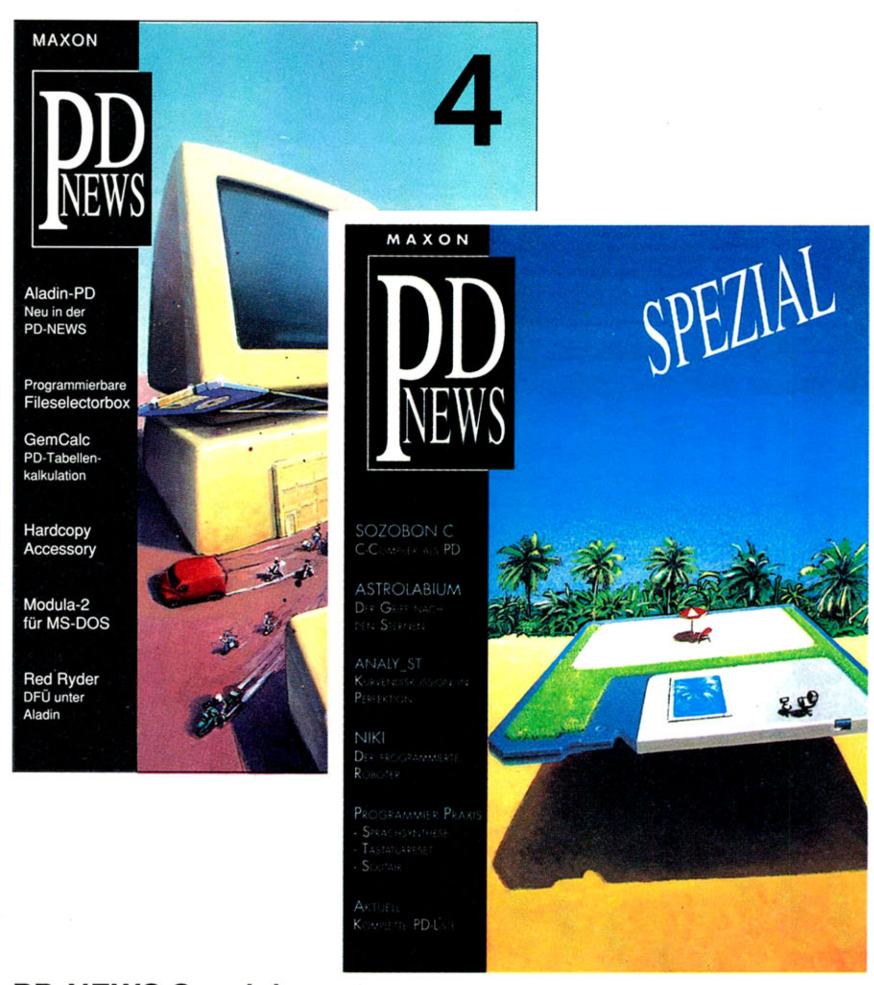


KNIFFEL: Flotte Kniffelversion, mit allem, was dazugehört (Kleine Straße, Große Straße, 3er Pasch, Chance...)

Zeichenerklärung:
s/w = nur monochrom
f = nur Farbe
1MB = mind. 1MB freier Speicher nötig

Machen Sie mit!

Möchten Sie ein selbstgeschriebenes Programm in unsere PD-Sammlung geben, um es auch anderen Usern zugänglich zu machen? Kein Problem. Schicken Sie es uns auf einer Diskette zu, samt einer Bestätigung, daß es von Ihnen geschrieben wurde und frei von Rechten Dritter ist. Bei Fragen steht Ihnen die Redaktion gerne zur Verfügung.



PD-NEWS Spezial

Das erste Sonderheft ist da. Der Schwerpunkt liegt im wissenschaftlichen Bereich, wo der ST sehr stark vertreten ist. Aber auch andere wichtige Programme werden vorgestellt und mit vielen Tips&Tricks bedacht. In der Programmierpraxis lernt Ihr ST das Sprechen, aber lesen Sie doch selbst in der neuen PD-NEWS SPEZIAL.

Aus dem Inhalt:

SOBOZON C: Ein C-Compilersystem für den ST
ASTROLABIUM: Der Griff nach den Sternen
ANALY_ST: Kurvendiskussion in Perfektion
NIKI: Der programmierte Roboter
SPEKTREN: Massenspektrometrie
REGELSIM: Regelungstechnik
FONTMASTER: Der Druckgigant
ST:KLICK: Multiaccessory

Programmierpraxis
SOLITÄR: Die errechnete Lösung
SAM: Sprachsynthese im ST
TASTATURRESET: Vierfingergriff

Als besondere Dreingabe enthält die PD-NEWS die komplette Liste aller PD-Disketten von 1-245.

Die PD-NEWS wird bei jeder Bestellung (solange Vorrat reicht) kostenlos beigelegt. Einzelbestellungen gegen frankierten Rückumschlag (DM 2.40, DIN A4) möglich.

BITTE BEACHTEN

Sämtliche Disketten können ab dem Erstverkaufstag der ST-Computer direkt bei der MAXON-Computer bezogen werden.

Bitte beachten Sie folgende Punkte:

1. Schriftliche Bestellung

- Der Unkostenbeitrag für eine Diskette beträgt DM 10,-
- Hinzu kommen Versandkosten von DM 5,- (Ausland DM 10.-)
- Bezahlung nur per Scheck oder Nachnahme
- (Im Ausland **nur** Vorrauskasse möglich)
- **Ab 5 Disketten entfallen die Versandkosten (DM 5.- bzw. DM 10.-)**
- Bei Nachnahme zuzüglich DM 4,00 Nachnahmegebühr
- **Jeder Bestellung liegt (solange Vorrat reicht) die neuste PD-NEWS bei.**
- Einfacher geht's nicht
  Nutzen Sie die PD-Karte in diesem Heft

Bitte fügen Sie keine anderen Bestellungen oder Anfragen bei!

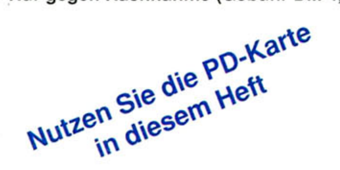
Adresse:

MAXON-Computer GmbH
'PD ST-Computer'
Postfach 5969
D-6236 Eschborn

2. Anruf genügt

MAXON-Computer GmbH
'PD-Versand'
Tel.: 0 61 96 / 48 18 11
Mo-Fr 9^{00} - 13^{00} und 14^{00} -17^{00} Uhr

- Nur gegen Nachnahme (Gebühr DM 4,00)



WANTED!

WORDPLUS-Treiber gesucht

Wir möchten demnächst eine neue PD-Diskette mit Druckertreibern für WORDPLUS zusammenstellen, da viele User mit dem Druckertreiber Probleme haben, man denke nur an die Installation. Wenn auch Sie einen eigenen, ausgereiften Treiber erstellt haben, und ihn auch anderen geplagten Usern zur Verfügung stellen wollen, dann senden Sie ihn bitte an uns ein. Enthalten sein sollten jeweils die installierte CFG-Datei, nach Möglichkeit das zugehörige HEX-File sowie eine kurze Beschreibung, welche besonderen Kommandos eingebaut wurden.

SIGNUM!- und GEM-Fonts gesucht

Auch im Bereich der Zeichensätze werden neue PD-Disketten erstellt. Sollten Sie eigene erstellt haben, die Sie als Public Domain freigeben wollen, schicken Sie sie uns bitte zu. Zur Aufnahme in die PD-Sammlung benötigen wir eine Bestätigung, daß der Font von Ihnen erstellt wurde und frei von Rechten Dritter ist. Bitte schicken Sie nur eigene Fonts, da wir nur diese aufnehmen können.

MAXON Computer
PD-Einsendung
'Aktion Druckertreiber'
'Aktion FONTS'
Industriestr. 26
6236 Eschborn



Sonderdisk

Auch diesmal begrüßen wir Sie wieder zu unserer Sonderdisketten-Sammlung. Für alle Spielefreunde ist eine Handelssimulation hinzugekommen, die neben durchdachten spielerischen Elementen auch durch schöne Grafik besticht. Bitte denken Sie daran, daß Sonderdisks keine PD-Disketten sind. Die Versandbedingungen entsprechen mit Ausnahme der DM 15,- denen der PD-Disketten.

TOS 1.0

Hier handelt es sich um die Diskettenversion des TOS von 1986, die ein wenig anders ist als die aktuelle Version. Es kann vorkommen, daß ältere Programme nicht mehr mit dem aktuellen Betriebssystem funktionieren. Was tun? Einfach die TOS-Diskette einlegen, den Rechner starten und schon ist alles beim alten.



TOS 1.0 DM 15,-

RCS

Das Resource Construction Set ist ein sinnvolles Utility, womit man ganz einfach Menüleisten und Dialogboxen für eigene Programme erstellen kann. Es ist ein Muß für jeden Programmierer, der seine Software professionell gestalten möchte.



RCS DM 15,-

Extended VT52-Emulator

Schneller als der Blitter
Dieses Utility ersetzt den im TOS integrierten VT52-Emulator vollkommen. Es enthält neue Routinen zur Bildschirmausgabe, die wesentlich schneller sind als die im TOS eingebauten. Daher erfolgt eine Beschleunigung der Textausgabe um den Faktor 3 bis 5 je nach Komplexität. Neben der Beschleunigung verfügt der Emulator über zusätzliche, programmierbare ESC-Funktionen, wie z.B. Softscrolling nach allen vier Seiten.



VT52-Emulator DM 15,-

Accessories
Kleine, große Helfer I

LOVELY HELPER
... ist ein Accessory, das sechs kleine, große Programme umfaßt: Einen Taschenrechner, einen Kalender, den man auf jedes Bundesland abstimmen kann, eine Uhr, die in die Menüleiste eingeblendet wird, eine Directory-Anzeige und -Druckoption, die nach belieibigem Kriterium den Inhalt einer Diskette ausdruckt und einen komfortablen Drucker-Spooler. Der Quellcode liegt in Pascal bei.



Accessories I
Lovely Helper DM 15,-

Accessories
Kleine, große Helfer II

PARTCOPY
Dieses Programm erlaubt Ihnen, einen Teil des Bildschirms abzuschneiden und auf dem Drucker auszugeben oder auf Diskette zu speichern. Zum Ausschneiden stehen Ihnen verschiedene Möglichkeiten zur Auswahl: rechteckig, polygonal und mit einem Lasso. Der Quellcode liegt in Assembler bei.

SPOOLER
Drucken, aber trotzdem mit dem Rechner weiterarbeiten zu können, das ist heutzutage ein Muß. Was das ATARI ST-Betriebssystem nicht kann, bietet dieser schnelle Spooler. Der hier vorgestellte Spooler als Accessory kann nicht nur Texte, sondern auch Hardcopies 'spoolen'. Abarbeitungsgeschwindigkeit und Größe des reservierten Speicherplatzes können Sie selbst einstellen. Der Quellcode liegt in Assembler bei.

PRINTTOP
Möchten Sie eine Hardcopy von nur einem bestimmten Fenster machen? Vielleicht trifft das bei Ihnen des öfteren zu. Mit diesem Accessory werden Sie immer in der Lage sein, das gerade aktive Fenster auf einem Drucker auszugeben. Der Quellcode liegt in Modula-2 bei.



Accessories II
Partcopy, Spooler & Printtop
DM 15,-

NIKI - der Roboter

Das Programm NIKI ist eine geschlossene Programmierumgebung mit eigenem Compiler und Editor für den Anfangsunterricht in PASCAL. NIKI ist das Modell eines programmierbaren Roboters, der sich auf einem Arbeitsfeld der Größe 10*15 bewegen und dort Gegenstände aufnehmen und ablegen kann. Roboterprogramme sind gewöhnliche, PASCAL-ähnliche Programme mit allen Kontrollstrukturen (Verzweigungen und Schleifen), aber ohne Variable. Mit etwas Phantasie lassen sich recht komplexe Aufgaben lösen. Ein Teach-in-Modus ist ebenso vorhanden. Dadurch 'lernt' unser kleiner Roboter, bestimmte Dinge auszuführen. Eine ungewöhnliche Hilfe zum Erlernen der Programmiersprache Pascal!



NIKI DM 15,-

VIRUS EX

VIRUS EX dient der Früherkennung von Viren. Es prüft Programme auf Veränderungen und erkennt somit Viren an der Verbreitung. Damit ist VIRUS EX bereits für die Zukunft programmiert! Auch eine Diskette oder Partition kann untersucht werden. Zwei andere Programme zur Viruserkennung und -bekämpfung sind Bestandteil dieser Diskette:

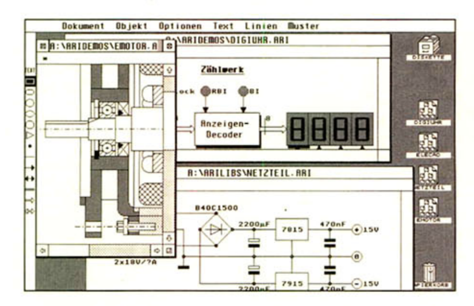
BCH
Der BootsektorCHecker wird in den AUTO-Ordner kopiert. Er meldet dann bei jedem Bootvorgang, ob der Bootsektor ausführbar ist oder nicht.

SWatch
SWatch ist ein kleines Accessory, das regelmäßig den Speicher überwacht (resetfeste Programme, Systemvariablen, VBL-Queue) und den internen Diskettenpuffer auf einen ausführbaren Bootsektor prüft.



VIRUS EX DM 15,-

ARIADNE



ARIADNE ist ein besonderes Zeichenprogramm. Es ist objektorientiert und bietet die Möglichkeit, jedes beliebige Grafikobjekt (mit Doppelklick) zu öffnen, worauf eine neue Zeichenebene bereitgestellt wird. Die Objekte auf dieser Ebene können dann wiederum geöffnet werden usf. Diese hierarchische Struktur eignet sich besonders zur Darstellung komplizierterer Dinge, z.B. Blockschaltbilder, Schaltungen, Software-Dokumentation. Voll in GEM eingebunden und sehr leicht zu erlernen. Auch als Zeichenprogramm für Studien- und Diplomarbeiten o.ä. (als Hardcopy in Signum!) geeeignet.

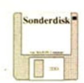


ARIADNE DM 15,-

LEGENDE



Legende ist ein Spiel für zwei bis vier Personen. Es geht in erste Linie darum, ein Königreich erfolgreich zu verwalten, ohne zu vergessen, daß man den Bewohnern gegenüber gewisse Verpflichtungen hat. Unvermeidlich sind die militärischen Aktionen gegen andere Königreiche, deren Ländereien oder Burgen Sie plündern wollen, oder wenn Sie von einem expansionistischen Trieb befallen sind. Sie sollten aber auch das Wirtschaftliche nicht vergessen, denn nur damit können Sie Ihre Mitspieler besiegen. Am Ende muß nämlich ein einziges Königreich übrigbleiben. (f, 1MB)



LEGENDE DM 15,-

Immer up to date

Mit dieser Sparte wollen wir allen unseren Lesern die Möglichkeit geben, sich über die neuesten Programm-Versionen zu informieren. Angegeben werden die aktuelle Versionsnummer, ein eventueller Kopierschutz, die Bildschirmauflösungen und der Speicherbedarf. Softwarefirmen ist es somit möglich, die ST-Computer-Leser über ihre Updates zu informieren.

Programmname	Version	Daten	Programmname	Version	Daten
Adimens ST	3.0	N HM	Mega Paint	2.20	N H 1M
Aditalk ST	2.3	N HM	Megamax Modula 2	3.5	N HM
Adress ST / Check ST	1.0	N H	Micro C-Shell	2.70	N HM
Afusoft Morse-Tutor	2.0	N HML	MT C-Shell	1.20	N HM 1M
Afusoft Radio-Writer	1.0	N HML	Multidesk	1.82	N HML
Afusoft Radiofax plus	1.0	N HML 1M	Musix32	1.01	J H
Aladin	3.0	J H	NeoDesk	2.0	N HML
AnsiTerm	1.4	N	Omikron Assembler	1.86	N HML
Assembler Tutorials	1.05	N	Omikron BASIC Compiler	3.06	N HML
Banktransfer	1.0	N H	Omikron BASIC 68881-Compiler	3.06	N HML
1st BASIC Tool	1.1	N HML	Omikron BASIC Interpreter	3.02	N HML
BTX/VTX-Manager	3.0	N H 1M	Omikron DRAW! 3.0	3.01	N HML
Calamus	1.09.2	N H 1M	Omikron EasyGEM-Lib	1.0	N HML
Cashflow	1.0	N H 1M	Omikron Maskeneditor	1.0	N HML
CIS-L&G	1.01		Omikron Midi-Lib	2.1	N HML
Creator	1.0	N H	Omikron Numerik-Lib	1.2	N HML
dBMAN	5.10	N HML	Omikron Statistik-Lib	1.5	N HML
fibuMAN	3.0	N H	PAM's TERM/4014	3.012	N H
fibuSTAT	2.3	N H	PAM's TurboDisk	1.7	N HML
Flash-Cache/Flash-Bak	1.0	N HM	PAM's NET	1.0	N HML
Flexdisk	1.3	N HML	PCB-layout	1.19	N H
FM-Meßtechnik	1.0.b	N HM	PC ditto Euroversion	3.96	N HML
Gadget	1.2.5b	N H	PegaFakt	1.3	N H
GEMinterface ST	1.1.	N HML	phs-BTX-Box	6.0	N HML 1M
GFA-Artist	1.0	N L	phs-ST-Box	1.2	N HM
GFA-Assembler	1.2	N HML	phs-Boxtalk	1.0	N HM 1M
GFA-BASIC 68881	1.3	N HML	phs-Boxedi	1.0	N HML 1M
GFA-BASIC-Compiler	3.02	N HML	phs-Cheapnet	1.2	N HM
GFA-BASIC-Interpreter	3.07D	N HML	Pro Sound Designer	1.2	L
GFA-Draft	2.1	N	Pro Sprite Designer	1.0	L
GFA-Draft plus	3.0	N	Protos	1.1	N H 1M
GFA-Farb-Konverter	1.2	N H	Revolver	1.1	N HML 1M
GFA-Monochrom-Konverter	1.2	N ML	Search!	2.0	N HM
GFA-Objekt	1.2	N HM	Signum! zwei	1.0	N H
GFA-Starter	1.1	N HML	Soundmachine ST	1.0	N HM
GFA-Vektor	1.0	N	SoundMerlin	1.0	N HM
G+Plus	1.4	N HML	SPC-Modula-2	1.42	N HML
GrafStar	1.0	N H	Spectre 128	1.9	J HM
Hänisch Modula-2	N.1	N HML	1st_Speeder 2	1.0	N HML 1M
Hard Disk Accelerator	1.0	N HML	STAD	1.3+	N H
Hard Disk Toolkit	1.05	N HM	Steuer-Tax 2.9	1.10	N HM
Harddisk Utility	2.2	N HM	Steuer-Tax 3.9	1.10	N HM
Imagic	1.1	N HML	STop	1.1	N HM
Intelligent Spooler	1.10	N HML	ST Pascal plus	2.06	N HM
Interlink ST	1.89	N HM	SuperScore	1.4	J H 1M
K-Resource	2.0	N HM	Tempus	2.05	N HM
Kleisterscheibe	2.0	N HM	TIM	1.2	N H
Label ST	1.0	N HML	TIM II	1.0	N H 1M
Laser C (Megamax)	2.1	N HML	Transfile ST 1600	1.1	N HM
1st_Lektor	1.2	HM	Transfile ST 850	1.1	N HM
Lern ST	1.22	N HML	Transfile ST plus	3.0	N HM
Link_it GFA	1.1	N HML	Turbo C	1.1	N HM
Link_it Omikron	2.0	N HML	Turbo ST	1.6	N HML
MagicBOX ST	7.58	N H 1M	VSH Manager	2.01	N HML 1M

Irrtum vorbehalten!
Daten-Legende : N = kein Kopierschutz, J = Kopierschutz, H = hohe Auflösung, M = mittlere Auflösung, L = niedrige Auflösung, 1M = mindestens 1 Megabyte

INSERENTENVERZEICHNIS

Computer & Electronic & Zubehör H E R G E S
Obere Rischbachstr. 88 · 6670 St. Ingbert
Telefon (0 68 94) 38 31 78 · Telefax (0 68 94) 38 28 55

Atari-Produkte:
Mega-ST 4, SM-124, Maus, Blitter, Omikron oder GFA kpl. 3398,-
Mega-ST 2, SM-124, Maus, Blitter, Omikron oder GFA kpl. 2338,-
Mega-ST 2, wie oben, zusätzlich auf 4-MB aufger. kpl. 3198,-
Mega-ST 1, SM-124, Maus, Blitter, Omikron oder GFA kpl. 1728,-
1040-STf, SM-124, Maus, Modulator, Omikron oder GFA kpl. 1328,-
Alle Atari-STs wie oben, nur ohne Monitor SM-124 abzügl. 330,-
Alle Atari-STs wie oben, nur mit SC-1224 Colorm. Aufpr. 666,-
Atari Festplatte Megafile 30 kompl. mit Zubehör kpl. 948,-
Atari-Festplatte Megafile 44 kompl. mit Zubehör kpl. 2222,-
SyQuest SQ-400 Wechselcartridge 44-MB kpl. 278,-
Atari Laserdrucker SLM-804 incl. Zub. + Interface kpl. 2800,-
Toner-Cartridge für Atari SLM-804 Laser kpl. 95,-
Atari SM-124 Monochrom-Monitor kpl. 370,-
Atari SC-1224 Color-Monitor kpl. 688,-
Orig. Megatastatur für Ersatz, Umbau, Zweittastatur kpl. 328,-
1st-WordPlus und 1st-Mail incl. deutsche Anleit. kpl. 138,-

Atari Ersatz-Chip's:
YM-2149F = 29,- Glue = 143,- MFP = 32,- Shifter = 175,-
WDC 1772-2 = 39,- MMU = 149,- Blitter = 148,- DMA = 151,-
Alle anderen Atari-Ersatzteile/Erweiterungen auf tel. Anfrage!

ST-Floppy's:
Floppy 3.5-T1, 3,5 Zoll, 720-KB formatiert kpl. 222,-
Floppy 5.25-T1, 5,25 Zoll, 720/360-KB formatiert kpl. 282,-
Floppy 5.25-T2, wie oben, kein Headload-Geräusch, Motor-ON nur bei Select, Led's f. EIN/AUS + 40/80 Tr. kpl. 333,-
Teac-Laufw. FD-55-FR/GFR incl. Atari-Modifikation kpl. 235,-
Teac-Laufw. FD-135/235-FN/HFN incl. Atari-Modifik. kpl. 185,-
Gehäuse 5.25 für FD-55-FR oder comp. in Abmessung kpl. 24,-
Gehäuse 3.5 für FD135/235 oder comp. in Abmessung kpl. 12,-
Gehäuse 5.25 wie oben + Netzteil für 5 + 12 Volt kpl. 55,-
Gehäuse 3.5 wie oben + Netzteil für 5 Volt kpl. 44,-
34-poliger Shugartbus-Stecker 5.25/3.5 Stck 3,-
Orig. Floppyanschlußkabel 1.8 m 2★14pol. Stecker Stck 18,-
4-poliger Spannungsstecker 3.5/5.25 Stck 3,-
Kippschalter 1★UM für Zentralbefestigung Stck 1,65
Buskabel Atari-ST auf Shugart, 5.25/3.5 Stck 22,-

Dyn. Rambausteine +:
DRAM 51.1000-70ns (1MB★1) 1. Wahl Marke Siemens Stck 38,-
DRAM's 51.1000-XXXns, 41.256-XXXns (XXX = 70-150ns) Stck tel. A.
Alle anderen Ram's verschiedener Hersteller Stck tel. A.

IC-Sockel 16-40pol. Low-Cost:
40p. = 1.00/28p. = 0.75/24p. = 0.68/20p. = 0.55/18p. = 0.50/16p. = 0.45

Netzteile für Eigenbau/Bastler:
Steckernetzteil 1amp. 1.5-12Volt schaltbar Stck 18,-
Schaltnetzteilplatine in Industriequalität, Abm. 160*100*45 mm, +5V/5amp, +12V/2.5amp,-12V/0.5amp Stck 79,-

Gleichrichter, Regler, usw:
Lüfter 80*80*25, Alu-Gehäuse, 220 Volt Stck 16,-
UA-7805, UA-7812, Belastung 1.5amp. gekühlt Stck 0,80
Rundgleichrichter B250/C800 Stck 0,80
Sieb-Elko liegend 1000yf/25Volt Stck 1,40
Sicherungshalter (Zentralbefestigung) Stck 1,10

Star-Drucker + Zubehör:
Star LC-10 original deutsche Ware incl. Druckerk. kpl. 478,-
Star LC-10 color deutsche Ware incl. Druckerkabel kpl. 628,-
Star LC-24/10 deutsche Ware incl. Druckerkabel kpl. 748,-
Vollautom. Blatteinzug für Star LC-10/LC-10Color kpl. 188,-
Farbband für Star LC-10 Color (4-farbig) Stck 16,-
Farbband für Star LC-10 (schwarz) Stck 8,-
Farbband für Star LC-24/10 (schwarz) Stck 12,-
Centronic's Drucker-Umschalter 2-fach Stck 69,-
Centronic's Drucker-Umschalter 3-fach Stck 89,-
Druckerkabel Atari-ST auf Centronic's Stck 15,-

Sonstiges für ST:
Omikronbasic V3.0 (Diskette + Handbuch) Stck 25,-
GFA-Basic V2.0 (Diskette + Handbuch) Stck 20,-
Dataphon s21d-2 incl. Atari-ST Schnittstellenk. kpl. 235,-
Atari-ST Floppyumschalter 3-fach (f. alle ST's) Stck 49,-
Floppy/-stecker/-buchse, Monitor/-stecker/-buchse Stck 4,-
Universal-Kleingehäuse (Plastic oder Metall) ab 8,-
GFA-Basic plus Compiler V2.0 incl. Handbuch Stck 44,-
Disk's 3.5/2DD NoName im 10erPack Top-Qualität Pack 18,-
Universal-Joystickverlängerung 3-Meter Stck 8,-

Sonstiges aller Art:
Autosound Autoradios, Autolautspr., usw.! Betreffendes Info anf.
Anrufbeantworter, Telefaxgeräte, Telefonzentralen! Betr. Info anf.
Telefone, Funkgeräte und Zubehör! Betreffendes Info anf.
Lieferung per UPS! Alle Preise in DM incl. MwsT, zzgl. Porto + Verp.!
★ Selbstabholung nur gegen Termin ★ Alle Angebote freibleibend

---

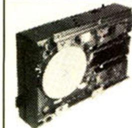
SciLab GmbH
Isestr. 57
2 Hamburg 13

ST-STATISTIK
- Univariate und multivariate Statistik
- Von Mittelwert bis Faktor-, Cluster-, Varianz- und Regressionsanalyse, medizinische Tests, T-Test u.v.a.
- Volle Grafikeinbindung in 2D & 3D
- Balken, Torten, Bänder, Linien und Dendrogramme mit dem Grafikeditor frei beschriften und gestalten
- Automatische (DIN) und wählbare Skalierung und Beschriftung d. Achsen
- Eigener Dateneditor, Ein-/Ausgabe in Textprogramme und Datenbanken
- Liest unbegrenzt große ASCII-Files
- Der Preis: DM 349.-

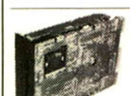
CHEMOGRAPH-PLUS
- Perfektes und schnelles Erstellen von Skizzen und Strukturformeln aus dem Bereich der Chemie in 2D, 3D und für stereographische Betrachtung
- Am Bildschirm editieren, rotieren, zoomen, spiegeln, plazieren etc.
- Nachladbare und erweiterbare Formelbibliotheken, wahlweise Normierung von Linien und Winkeln
- Komfortable Beschriftungsmöglichkeit
- Publikationsreife Metafile-Grafik mit Fonts und Treibern für alle gängigen Drucker (9/24 Nadler u. Atari-Laser)
- Der Preis: DM 699.-

Hotline 13[00] bis 16[00] 040/460 37 02

Laufwerke zu Sonderpreisen!



BASF-CANON-5¼"-DISKETTEN-LAUFWERK
Typ: AG 6138,
80 Spuren, Shugart-Datenbus, doppelseitig, High-Density-Betrieb im IBM-AT möglich. Paßt auch an den ATARI-ST oder AMIGA. Formatiert 720 KB. 360 KB möglich.
Slimline mit Blende nur DM **99.–**
dto., volle Bauhöhe nur DM **99.–**



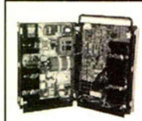
MINISCRIBE-FESTPLATTE 10 MByte
MFM-Codierung, 612 Datenzylinder, 2 Köpfe. Läuft mit jedem PC-Festplattencontroller, der diese Daten unterstützt.
Slimline mit Blende nur DM **199.–**
dto., volle Bauhöhe nur DM **199.–**
dto., wie vor, jedoch mit Controller für IBM-komp. XT-Rechner nur DM **399.–**

AKUSTIK-KOPPLER

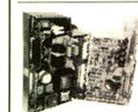


Fabr. Anderson-Jacobson, California. Mit FTZ-Nummer, komplett im Gehäuse, 220-VW-Anschluß. Standardschnittstelle: V.24/RS232. Betrieb: Halb- od. Vollduplex, seriell, binär, asynchron, 300 Baud. Übertragung: Frequenzmodulation. Größe: 11 × 19 × 32 cm, Gewicht: 2,3 kg. Einschl. Unterlagen
nur DM **100.–**
Passendes V.24-Kabel dazu DM **19.–**

COMPUTER-NETZTEILE



Primär getaktetes Netzteil für Computer usw.
Im aufklappbaren Metallgehäuse. Primär: 220 VW, 5 V=/100 A, 24 V=/10 A, –12 V=/5 A
DM **35.–**

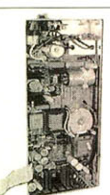


Primär getaktetes Netzteil
Im aufklappbaren Metallgehäuse. Primär: 220 VW, sek.: 5 V=/5 A, 12 V=/8 A, –12 V=/0,5 A, 24 V=/8 A, mit Anschlußplan
DM **25.–**

ACHTUNG! Alle hier aufgeführten Netzteile sind wenig gebraucht, jedoch alle einwandfrei und funktionsgeprüft!



Netzteil für Computer usw. Typ CNT 3
Flaches perforiertes Metallgehäuse mit Ringkerntrafo. Maße: 33,5 × 15 × 5 cm. Primär: 220 VW, sek.: 5 V=/1 A (4 A), 12 V=/2,5 A, 27 V=/2 A. Die 5-V-Spannung kann durch Auswechseln einer Sicherung und eines Spannungsreglers auf 4 A erhöht werden. Mit Unterlagen DM **26.–**



Primär getaktetes Netzteil Typ CNT 4
Flaches perforiertes Metallgehäuse. Maße: 33,5 × 15 × 5 cm. Primär: 220 VW, sek.: 5 V=/1 A, 5 V=/3 A, 12 V=/0,5 A, 12 V=/4 A, 27 V=/2 A. Mit Unterlagen DM **26.–**



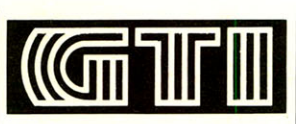
ATARI ST an NEC MULTISYNC
und kompatible Monitore. Umschalter für Monochrom und Colormodus an einem ST. Audio-Ausgang separat vorhanden. Abgeschirmtes Stahlblechgehäuse. Softwaremäßige Umschaltung ist möglich
nur DM **49.–**



DRUCKERKABEL,
ca. 1,9 m lang, für IBM und kompatible Rechner, ATARI ST und AMIGA. SUB-D 25pol., an Centronics 36pol. DM **9,95**

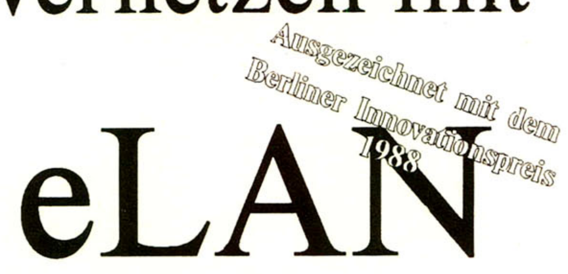
NADLER electronic

Ladenverkauf: 4600 Dortmund, Bornstr. 22
Tel. (0231) 52 30 60, Fax (0231) 57 25 60
3000 Hannover, Herschelstraße 31
Tel. (0511) 32 63 61, Fax (0511) 13 450
Angebot freibleibend. Ab Dortmund. Versand per Nachn. nicht unter DM 20,- (Ausland DM 50,-)



Systemlösungen für die Qualitätssicherung
Software · Hardware

In Zukunft Vernetzen mit



Das Netzwerk für ATARI ST- Computer, das schon wieder einen Schritt weiter ist

eLAN-TOS 2.0

eLAN-MEGA
eLAN-DOS
eLAN-OS9
eLAN-ST
eLAN-PC
eLAN-VME
eLAN-ECB

Wir garantieren Sicherheit - mit eLAN !

GTI
Gesellschaft für technische Informatik mbH • Berlin
Unter den Eichen 108a
1000 Berlin 45
(030) 8 31 50 21/22

Kleinanzeigen

BIETE HARDWARE

Atari-Blitter-ROM-TOS, a. m. IBM- Grafikzeichen, 100 DM, 02630/7525

Verk. **Adv. Corruption 40,** Drucker **Scanner 100,** 02368/55020

Atari 520+, SF314, SM124, Maus, **mit Originalsoftware** (Laser C, Star Writer ST, Profi Painter) DM 1150,- Tel. 08363/8891 ab 18 Uhr

Mega ST 4 + Harddisk SH 204 + 1 ext. Floppy 3,5 Zoll + 1 ext. Floppy 5 1/4 Zoll + EPSON Farbdrucker JX80 NLQ + div. Büchern und Disketten VB 5000 DM Tel. 040 647 67 25

32MB-HD VB DM 750 02237/4880

Speichererw. f. Atari 520,260 ST auf 1 MB m. Einbau 260 DM. Atari ST auf 2,5 MB 860 DM. Bausatz auf 1 MB 220 DM, auf 2,5 MB 700 DM. Tel. 069/686491

Laufend neu + orig. Tastat. + Gehäuse + Laufw. Rauch 07031/52607

Speichererw. für 260/520 ST auf 1 MB 290 DM. 520/1040 auf 2,5 MB 950 DM. Tel. 08856/7287

520ST+ SFr. 500,-; ST25/ SFr 350,-; Omti 5510 SFr 100,-; ST-01 SFr 50,-; Adaptec 4000 SFr 200,-; Aladin V3.0 mit ROMs SFr 350,-; P. Gilli; Wickenweg 15; CH-8048 Zürich

Sampler 8 Bit 110 DM 02366/41014

1040STF, SM124 + 3,5" Laufwerk, 10 Mon. alt. VB 1000. Tel. 0911-513717

Maus zu verk. DM 48 02291/4919

PC-SPEED für ATARI ST
zum günstigen Einführungspreis
Neuendorf Computersysteme
Tel.: 0201-408184

Festplatte 105MB 18MS SCSI mit Uhr anschl.fertig FP 1900,- DM/ Mega 2 im AT-Gehäuse mit 105MB Platte, 5,25" + 3,5" LW, SM124 FP 4800 DM; 3,5" LW 720 KB neu 180 DM uvm! Tel. 02041-64204 ab 19[00]

Signum 378,00 / Harddisk 30 MB 40ms 1198 DM / Omti 5520 214 DM / Sybex-Elektor Bücher LQ400 830 DM
Hoffmann Elektronik 0833186371

Blitter-TOS o. altes TOS + TOS 1.4 NEU gleichzeitig in allen ST, umfangr. Anleit. K. Ratsch, Herner Str. 127, 4350 Recklinghausen, 02361/28442 ab 19.00

Atari-Blitter-Chip 06281-8177

Vortex HDPL30 950,- 030-249409

Festplatte SH205 07121/600335

BIETE SOFTWARE

Public Domain Software
für alle
Atari ST-Modelle

Public Domain Software
sollte nicht nur Spaß machen –
sondern auch für's Taschengeld
erschwinglich sein!
Wer nicht für
JEDES BIT ZAHLT,
es aber haben möchte, schreibt
uns!
Klaus Kohler
Don-Carlos-Str. 33 B
7 Stuttgart 80

aladin
Filetransfer nach TOS
mit Konv. v. Text u. Grafik
nach Degas, STAD u. IMG 59,-
M. Nieß, Am Kottenberg 34
4799 Borchen, 05251/391849

PD-SOFTWARE
Alle ST-Disks SS je DM 4,–
(incl. 2DD-Diskette!)
Doppelseitig DS je DM 6,–
Versand: bei Scheck DM 3,–
bei Nachnahme DM 8,–
N. Twardoch, Gröchteweg 22
4902 Bad Salzuflen 1

★★PD für ATARI-ST, IBM★★
Atari(650), IBM(2700), **3–6 DM**
Kat: PC-10DM (7 Disk.), ST-5DM
Johrend, Neusalzer Str. 9, 85 Nbg

Anwender- u. Spielesoftware
Laufend n. Programme a. Lager
Riesenauswahl an Original
Soft- u. Hardware
Gratiskatalog anf. o. anrufen
W. Wünsch
Soft- u. Hardware,
07231/766595
Friedenstr. 212, 7530 Pforzheim

SOFTWAREBERATUNGS - ENTWICKLUNG
EDV - Profi berät Sie und entwickelt Software speziell für Ihre Bedürfnisse.
Info: B. Noack, v.- Eichendorff Str. 6, 8015 Markt-Schwaben.
Tel.: 0 81 21/ 59 32

Softstation
2 PD's von ST-Comp. Fortl. Nr. (1–2 usw.) DM 5,–
Bel. Komb. DM 7,–
sonstige Serien DM 7,–
Disketten wahlw. blau, weiß, rot, grün, gelb od. orange
Verp. Porto plus NN DM 7,–
Tel. 07195-53707

Public-Domain-Blitzversand!
Riesenauswahl! Preise:
ab DM 4,– einseitige Disketten
ab DM 6,– doppelseitige
Inclusive Diskette!!!
Auch alle ST-Disks! Auf Doppeldisks beliebig kombinierbar!
Gratisliste anfordern bei:
A. Gauger Software
Buhlstraße 16a, 7505 Ettlingen
0 72 43 / 3 18 28
Bitte Computertyp angeben!!

LERNPROGRAMME
1.) PD-Programme
2.) Kommerzielle Programme
LTA LERNSOFT 089/8545531
8032 Lochham, Friedenstr. 7

Leonardo Spitzen-Grafik-Prg. für **Atari-ST** monochr. 2 Bildschirme Füll-Muster-Editor, Überblenden umfangr. Optionen ü. Pull-Down. **DM 59,-** priv. Tel: 07221-53306

Räumungsverkauf
Lagerräumung f. neue Software
Solange Vorrat reicht.
Liste anf. o. gleich bestellen
W. Wünsch
Soft- u. Hardware
07231/766595
Friedenstr. 212, 7530 Pforzheim

Public-Domain-Software
Preis je 2DD-Disk DM 4 bis 5
Hard- u. Software zu SUPER
PREISEN. Kostenloses INFO
bei: B. Jürgensen Hard&Soft
Holmberg 4 · 2398 Harrislee

★ ATARI ST SOFTWARE ★
Einnahmen-Überschuss-Prog. 99,-
Diskettenverwaltung 49,-
Gratisinfo bei: Thorsten Lavid
Webschulstr. 44, 4050 M'Gladbach 1

CCS COMPUTER SHOP
Hard & Software – Ersatzteile
Markendisketten 3,5 10 St.
1D ab 20,- 2D ab 25,- DM bei
gr. Mengen Rabatt. Grafiken f.
Signum/STAD ca. 1500 20,-
Info anfordern: CSS Computer
Shop-Langenhorner Ch.670d
2 Hamburg 62.
Computer angeben.

PD-Software ab 2,95 DM incl. Disk. Liste gratis. Neumann Edingkweg 6, Dortmund 13, Tel. 215005

Wärmebedarfsberechnung DIN4701
KZahlberechnung DIN4108
Dampfdiffusion-Tauwasserschutz
Wärmeschutznachweis WäSchV
Dipl. Ing. V. Koch, Am Mehnacker 11
3563 Dautphetal 3. Tel. 064687652

Wärmebedarf DIN4701 ★ K-Zahl
★ Heizflächenauslegung ★ Rohrnetz ★ Demodisk 2-seitig für DM 10 Vorkasse von J. Binder, Eichendorffstr. 15 · 5030 Hürth

Grafiksammlung 25 Disk - Motive aller Art - Katalog DM 20,- wird mit Erstbestellung vergütet, je Disk DM 20,- Vorkasse-Nachname. Weindl Karl, Leutenbacherstr. 12, 7057 Nellmersbach

GRAU Graustufe für s/w-Bilder DM 40 + NN, M. Doerjer Kreuznacher Str. 12 · 6530 Bingen 17

Baustatik Baustatik Baustatik
FEM-Plattenprg, Eb. Stabwerkprg
Dipl.-Ing. U. Precht, Hüttenkamp 11, 4970 Bad Oeynhausen 1

Public-Domain-Software
Kopie auf unsere/Ihre, single/ double, Marken/NN-Disketten. Preise?? – kaum zu glauben!
Gratiskatalog bei: O. Schwede, Röntgenweg 9/1, 7050 Waiblingen

Original Ahnendatei V5.0 Wolfg. Pfeiffer, Espen 5, Dormagen 11

ST-Pascal + 120, MCC-Ass. 80, 1st Word + 60, Hanse 30, Profimat 40 etc. Tel. 09191/32623

SUCHE HARDWARE

Dringend preiswerte Speichererweiterung auf 1Mbyte für 260ST, Grauwertscanner für STAD bis 200,- DM und Star NL10

KONTAKTE

Softwareberatung/Entwicklg. EDV-Profi berät Sie und entwickelt Software speziell für Ihre Bedürfnisse. INFO: B. Noack v. Eichendorffstr. 6, 8015 Markt-schwaben; Tel: 082181/5932

★★ Softwareentwicklung ★★ Schreibe Software, die speziell auf Ihre Bedürfnisse zugeschnitten ist. Ks. Info anford. Thomas Leufkes, Knappenstr. 10, 4353 Oer-Erkenschwick. 02368/55020

VERSCHIEDENES

GEBURTSHOROSKOPE!
Exakte Zeichnung, Aspekte, Berechnungsdaten. Interpretation für gewerbl. Einsatz!
Tel. 06151/23720

ST-Computer Erstausgabe Feb.86– Jan.88 Topzust. z. verk. 09549/1340

★★★ SCAN-STUDIO ★★★
Wir scannen Ihre Vorlagen bis DIN A4, bis 600 dpi. Nur DM 3,– pro Vorlage. Auch Kleinstaufträge. Texterkennung möglich.
Alle Formate. Tel. 02451-68179

Suche ST Computer Sonderheft Nr. 2 (oder Kopie) 0731/34848

Suche Textaufnahme nach Diktat auf Atari ST. Wer schreibt medi-zin. Text (ca. 80 S.) Leicht (kein Ausdruck) nach Diktat! Preis ? Angebote an: H. Bresser, Schumacherring 27, 8000 München 83

THEMA "PUBLIC DOMAIN"**:

"Echt toll !!!"

Klaus M. ist begeistert. Wie viele andere User sucht er für seinen [Computer] große Software für wenig [Geld] und hat deshalb unseren PD-Software-[Katalog] gelesen.

Der [Katalog] hat es in sich! Alles über die Programme, geprüft ('Schrott' ist im [Papierkorb]), ausführlich erläutert und sortiert.

Und es gibt fast Alles: (z.B. Textverarb., Dateiverwalt., Tabellenkalk.,..), ABc (für SIGNUM!), (incl. DTP-Grafik-Lib.), nach (z.B. Modula-2, C.), (incl. DFÜ u. Acc.-Lib.), (Lernen/ Strategie/Aktion,..) und (incl. neuer Midi-Software.). Mit Klasse statt Masse.

Machen Sie's wie Klaus M.- (incl. Probe-*) für 5 (DM) oder bestellen!

* Bez. für frei (legal!) kopierbare Programme: "PD" ** Für 3,-DM bekommen Sie 'nur' den PD-Katalog

PD-EXPRESS
JÖRG RANGNOW SOFTWARE
ITTLINGER STR.45 7519 EPPINGEN-3
☎ 07262 / 5131 (AB 17.00 UHR)



Scanner/Printer DIN A4 "SPAT"

(Ausführlicher Testbericht in "ST-COMPUTER" 5/89)
Incl. Software und deutschen Handbuch. Zusätzlich mit einigen sinnvollen PD-Programmen.

DM 1.098,–– ab Werk

Wir unterhalten eine eigene Fachwerkstatt, und wir können eine Ersatzteilversorgung über Jahre hinaus garantieren!

Prospekt und Testbericht anfordern bei:

Ringhut GmbH Postfach 1645, 6070 Langen
Tel.: 06103/52660 Fax.: 06103/28555 Tx.: 17610392

ITF-32 I/O-INTERFACE

32 EINGÄNGE — 32 AUSGÄNGE
KASKADIERBAR
für ATARI - ST
ITF-32 ROM-PORT
I/O-INTERFACE
EINGÄNGE AUSGÄNGE
DM 187,-
KABELTESTER — ALARMANLAGE
RECHNERKOPPLUNG MIT 32 BIT

JÜRGEN SIEBER • RAFFELTERSTEIGE 15 • 7100 HEILBRONN • ☎ 07131/161151

Atari Bausätze und Bausteine

MMU	DM 124,30		AY-3-8910	DM 17,50
DMA	DM 124,60		WD1772	DM 29,00
Shifter	DM 133,30		68901	DM 23,00
GLUE	DM 113,60		68B50	DM 4,70
RP5C15	DM 25,00		68000PS	DM 16,80

PLATON (Version 1.3) Leiterplatten CAD	DM 298,00
Demo-Version	DM 20,00
511000-70	DM 29,90
41256-100	DM 9,90
414256-10	DM 32,00
1,2 Mb Disks im Atari-Laufwerk, ST 7-8/89 10.24 Mhz Quarz, 74 LS 157	DM 10,00
HYPER-TAST Interface für MF2-Tastatur Atari-kompatibel, keine Software erforderlich	DM 149,00
dto. Set mit Cherry-Tastatur, kpl.	DM 333,00
FD 1036 3,5" Lw., anschlußfertig, Netzteil	DM 220,00
Porto und Verpackung	DM 6,90

Zwischenverkauf und Irrtum vorbehalten

edicta Vertriebsgesellschaft für elektronische Bauelemente mbH
LÖWENSTRASSE 68, 7000 STUTTGART 70
TEL. 07 11/76 33 81, FAX 07 11/76 78 24

In der nächsten ST-Computer lesen Sie unter anderem

Script - Textverarbeitung à la Mac

Auf der ATARI-Messe in Düsseldorf wartete Application Systems mit einem neuen Textverarbeitungsprogramm namens Script auf. Wer sich ein bißchen auf dem Macintosh auskennt, weiß auch sofort, an welches Programm Script sich anlehnt, nämlich WriteNow, das auf dem Mac sehr verbreitet ist. Nach dem großen Erfolg von Signum! in der gehobeneren Anwendungsklasse will man nun in Heidelberg anscheinend auch Programmen wie Wordplus zu Leibe rücken. Man darf gespannt sein.

Modems und Mailboxprogramme

Die Datenfernübertragung gewinnt in der EDV einen immer höhreren Stellenwert. Welche Modems und Akustikkoppler Sie zur aktiven Teilnahme benötigen, wie gut sie sind und welche Mailboxprogramme welche Vor- und Nachteile haben, erfahren Sie in der Novemberausgabe.

Adimens ST plus

Bei den Datenbankanwendungen auf dem ATARI ST hat sich mit Sicherheit Adimens ST einen Namen gemacht. Nach der Netzwerkversion Adimens STI gibt es von ADI jetzt eine neue Adimens-Version, Adimens ST plus (3.0). U.a. sind Verbunddefinitionen wie Join und View implementiert worden.

Die nächste ST Computer erscheint am Fr., den 27.10.89

Fragen an die Redaktion

Ein Magazin wie die ST-Computer zu erstellen, kostet sehr viel Zeit und Mühe. Da wir ja weiterhin vorhaben, die Qualität zu steigern (ja, auch wenn das manchmal danebengeht), haben wir Redakteure ein großes Anliegen an Sie, liebe Leserinnen und Leser:

Bitte haben Sie Verständnis dafür, daß Fragen an die Redaktion nur Donnerstags von 14^{00}-17^{00} Uhr telefonisch beantwortet werden können.

Vielen Dank für Ihr Verständnis

Impressum ST Computer

Chefredakteur: Uwe Bärtels (UB)

Redaktion:
Uwe Bärtels (UB)
Harald Egel (HE)
Marcelo Merino (MM)
Harald Schneider (HS)
Martin Pittelkow (MP)

Redaktionelle Mitarbeiter:
C.Borgmeier (CBO), Fernando Brand (FB), Claus Brod (CB), Ingo Brümmer (IB), Stefan Höhn (SH), Raymund Hofmann (RH), Jürgen Leonhard (JL), Claus P. Lippert (CPL), Markus Nerding (MN), Chr. Schormann (CS), R.Tolksdorf (RT)

Autoren dieser Ausgabe:
D.Brockhaus, Dr.A.Ebeling, A.Esser, Dr.L.R.Frank, E.Grah, A.Hollmann, O.Krämer, D.Kühner, A.Lötscher, M.Malich, B.Malle, T.W.Müller, N.Preisler, D.Rabich, J.Stessun, T.Werner

Auslandskorrespondenz:
C.P.Lippert (Leitung), D.Dela Fuente (UK)
L.Hennelly (Nordamerika)

Redaktion: MAXON Computer GmbH
Postfach 59 69
Industriestr. 26
6236 Eschborn
Tel.: 0 61 96/48 18 14, FAX : 0 61 96/4 11 37

Verlag: Heim Fachverlag
Heidelberger Landstr. 194
6100 Darmstadt 13
Tel.: 0 61 51/5 60 57, FAX : 0 61 51/5 56 89 + 5 60 59

Verlagsleitung:
H.J.Heim

Anzeigenverkaufsleitung:
U.Heim

Anzeigenverkauf:
K.Margaritis

Anzeigenpreise:
nach Preisliste Nr.3, gültig ab 1.1.88
ISSN 0932-0385

Grafische Gestaltung:
Gabriele Imhof

Layout:
Kerstin Feist, Gülsu Okay, Manfred Zimmermann

Titelgestaltung:
Gunter Wenzel (Tel.: 06172/37193)

Fotografie:
Gabriele Imhof, Archiv, dpa

Produktion:
K.H.Hoffmann, B.Failer, S.Failer

Druck:
Ferling Druck W.Frotscher GmbH

Lektorat:
V.Pfeiffer

Bezugsmöglichkeiten:
ATARI-Fachhandel, Zeitschriftenhandel, Kauf- und Warenhäuser oder direkt beim Verlag

ST Computer erscheint 11 x im Jahr
Einzelpreis: DM 7,-, ÖS 56,-, SFr 7,-
Jahresabonnement: DM 70,-
Europ. Ausland : DM 90,- Luftpost : DM 120,-
In den Preisen sind die gesetzliche MWSt. und die Zustellgebühren enthalten.

Manuskripteinsendungen:
Programmlistings, Bauanleitungen und Manuskripte werden von der Redaktion gerne angenommen. Sie müssen frei von Rechten Dritter sein. Mit seiner Einsendung gibt der Verfasser die Zustimmung zum Abdruck und der Vervielfältigung auf Datenträgern der MAXON Computer GmbH. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte wird keine Haftung übernommen.



COMPUTER VIREN
Das neue
Anti Viren Kit 3.0
von G DATA
Kostenloser
Update-Service
für registrierte Benutzer.
Mit Harddisk-Überwachung.
Diagnose von Link-Viren.
Das meistverkaufte
Anti-Viren-Programm
für ATARI ST Computer
Anti Viren Kit 3.0
Anti Viren
DM 99,–
(unverbindl.
Preisempfehlung)
PROGRAMMABSTURZ!
DATENVERLUST!
Neue Version mit
Harddisk-Schreibüberwachung
und Virenscannung
Anti Viren Kit 3.0
bei Ihrem Atari-Händler
oder direkt bei
G DATA
Bundesrepublik:
Siemensstraße 16, D-4630 Bochum 1
Telefon. Bestellungen: 0 23 23/38 98 58
Versand zzgl. DM 5,– Portopauschale
per Nachnahme oder Vorausscheck.
Schweiz:
Computertrend, Langstr. 31, CH-8021 Zürich
Österreich:
CSR Computer Rittner, Hauptstraße 34,
A-7000 Eisenstadt